On the accuracy and efficiency of the reactor operation digital twin for parameter identification and state estimation[*]

Lizhan Hong,[1] Helin Gong,[1, †] Hongjun Ji,[1] Jialiang Lu,[1] Han Li,[1] and Qing Li[2]

[1]*Paris Elite Institute of Technology, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Science and Technology on Reactor System Design Technology Laboratory,*
Nuclear Power Institute of China, Chengdu 610041, China.

Accurate and efficient online parameter identification and state estimation are crucial for leveraging Digital Twin simulations to optimize the operation of near-carbon-free nuclear energy systems. In previous studies, we developed a reactor operation digital twin (RODT). However, non-differentiabilities and discontinuities arise when employing machine-learning-based surrogate forward models, challenging traditional gradient-based inverse methods and their variants. This study investigated deterministic and metaheuristic algorithms and developed hybrid algorithms to address these issues. An efficient modular RODT software framework that incorporates these methods into its post-evaluation module is presented for comprehensive comparison. The methods were rigorously assessed based on convergence profiles, stability with respect to noise, and computational performance. The numerical results show that the hybrid KNNLHS algorithm excels in real-time online applications, balancing accuracy and efficiency with a prediction error rate of only 1% and processing times of less than 0.1 s. Contrastingly, algorithms such as FSA, DE, and ADE, although slightly slower (approximately 1 s), demonstrated higher accuracy with a 0.3% relative L_2 error, which advances RODT methodologies to harness machine learning and system modeling for improved reactor monitoring, systematic diagnosis of off-normal events, and lifetime management strategies. The developed modular software and novel optimization methods presented offer pathways to realize the full potential of RODT for transforming energy engineering practices.

Keywords: Parameter Identification; State Estimation; Reactor Operation Digital Twin; Reduced Order Model; Inverse Problem

I. INTRODUCTION

A. Concept of Digital Twins (DTs)

The concept of digital twins (DTs) has gained significant traction in recent years, particularly in the engineering and industrial disciplines [1]. A DT refers to a virtual model that closely mirrors a real-world physical product, system, or process, serving as an effectively indistinguishable digital counterpart for practical purposes such as simulation, integration, testing, monitoring, and maintenance. DTs have been recognized as the fundamental premise of product lifecycle management, encompassing the entire lifecycle of the physical entity they represent, including creation, construction, operation/support, and disposal [2–6].

The value and fidelity of a DT representation depend on the specific use cases, considering that its granularity and complexity are tailored accordingly. Notably, a DT can be conceptualized and utilized even before its physical counterpart exists, allowing for comprehensive modeling and simulation of the intended entity's lifecycle.

The origin of the DT concept can be traced back to NASA, which pioneered its practical definition in 2010 as part of its efforts to enhance the simulation of spacecraft using physical model representations [2, 6]. Since then, the development of DTs has progressed in tandem with advances in product design and engineering. Traditional manual drafting techniques have given way to computer-aided drafting and design and model-based system engineering approaches, which establish a direct link between the DT and signals from its physical counterparts.

In recent years, DTs have garnered significant attention in various industrial sectors. Notably, their application in the chemical processing industry has demonstrated versatility [7]. In addition, DTs are increasingly utilized in simulation-based vehicle certification and fleet management [8], as well as in addressing the challenges of unpredictable and undesirable behaviors in complex systems [9, 10]. A specific area of growth has been observed in the nuclear plant industry, where digital twin architectures have been developed for the management and monitoring of nuclear plants, reflecting the expanding scope and potential of technology [11, 12].

B. Previous Work on Reactor Operation Digital Twins (RODTs)

A significant amount of research has been conducted to explore the potential applications of reactor operation digital twins (RODT) in the nuclear energy domain.

RODTs represent a specific instantiation of the DT concept, focusing on the numerical representation of nuclear reactors for real-time prediction, optimization, monitoring, control, and decision-making during their operational stage [13].

A previous study [13] introduced the first prototype of an RODT specifically tailored for the prediction of neutron flux and power fields in the operational stage of the HPR1000 reactor core [14]. The prototype demonstrated the feasibility of using the RODT for online monitoring and prediction of key reactor parameters, thereby enhancing operational understanding and performance assessment.

In [15], the forward model of the RODT was realized through the application of a nonintrusive reduced-order model constructed using an SVD autoencoder to learn the nonlinearity of the field distribution. Machine learning techniques, specifically leveraging k-nearest neighbour (KNN) and decision trees, are employed to establish forward mapping. Additionally, the inverse model adopts a generalized latent assimilation method for accurate estimation of the model parameters.

While demonstrating promise, opportunities remain for improving the efficacy of RODT for practical deployment. In particular, the inverse solver relies on a computationally expensive methodology that limits its potential integration within an online framework. Therefore, the aim of this work [16] is to enhance the key capabilities of the RODT. This includes proposing an advanced differential evolution algorithm to upgrade the inverse solver for improved efficiency and accuracy of parameter estimation. In addition, uncertainty quantification was conducted to validate the RODT's performance considering noisy observational data. Numerical validation experiments were performed across representative domains for an HPR1000 pressurized water reactor core to demonstrate its potential for engineering applications.

The key steps for constructing an RODT include: (i) training a reduced-order model (ROM) using model order reduction (MOR) [17–20] techniques as well as an efficient forward model and (ii) adaptation and implementation of inverse models to infer the input parameters and the resulting field distribution.

C. Introduction to Reduced Order Models (ROMs)

Many mathematical models encountered in real-life processes present computational challenges when employed in numerical simulations because of their inherent complexity and large dimensionality. To address these challenges, MOR techniques have been developed to reduce the computational complexity of these problems, particularly in the context of simulations involving large-scale dynamic and control systems. The ROM was computed as an approximation of the original model by reducing the dimensionality or degrees of freedom associated with the model.

The preparation of an ROM model involves two distinct phases, the offline and online phases, which are integral to the creation of a simplified representation of a high-fidelity full-order numerical model, enabling efficient real-time simulations and analysis. The offline phase focuses on determining the inherent low-dimensional structure of the underlying full-order model and deriving the reduced basis functions from high-fidelity snapshots of the system. Intrusive MOR techniques [21] are employed when a comprehensive understanding of the governing equations and numerical strategies employed in the full-order model is available. This requires access to the detailed numerical framework of the full-order model, which may involve proprietary commercial codes. By projecting a full-order model onto a reduced space, intrusive ROMs provide a concise representation with reduced computational complexity.

However, in practical engineering scenarios [22–24], detailed information regarding a full-order model is often unattainable, and the solvers or codes implementing these models are treated as black-box entities. This limitation poses a challenge to the implementation of traditional intrusive MOR methods. To address this issue, nonintrusive MOR techniques have been developed. Non-intrusive MOR approaches aim to construct ROMs without relying on detailed knowledge of the numerical framework of a full-order model. Instead, these methods establish an input-output mapping between the input parameters and the reduced basis through data-driven techniques such as interpolation, regression, or machine learning, approximating the behavior of the full-order model based on a reduced set of parameters and basis functions.

ROMs prepared through the offline and online phases play a critical role in the development of RODTs. RODTs require the integration of high-fidelity models and solvers with real-time capabilities, and ROM-based emulators provide promising solutions [21–23, 25]. Leveraging ROM-based forward solvers, RODTs can perform both one-to-one forward simulations and one-to-many simulations for inverse problems, encompassing parameter identification [26–29], data assimilation [24, 30–35], sensitivity/uncertainty quantification [36–40], and optimization/control [37, 41–47].

D. Forward Model Construction

In line with the framework RODT, as investigated in a previous study [13, 15], we selected KNN [48] to construct the forward model. The choice of KNN is justified by its numerous advantages, which render it a suitable candidate for the forward model in this context.

First, it is a nonparametric algorithm, meaning that it

does not assume any specific functional form for the relationship between the inputs and outputs. Second, KNN is relatively simple to implement and computes efficiently, particularly for low-dimensional input problems. The complexity of the algorithm depends primarily on the number of training samples, making it suitable for problems with moderate dataset sizes [49]. In a recent study [50], an investigation was conducted to explore the possibility of utilizing different values of K for each dataset, considering the information provided by the correlation matrix.

E. Inverse Model Construction

Generally, the inverse problem can be divided into two categories: (i) the first category involves solving the forward system with knowledge of an existing dataset of input and output; (ii) the second category involves solving the system's input with knowledge of the forward system and output [51]. In our case, the first category corresponds to the construction of the forward model, where we build a non-intrusive model as a surrogate to the full-order system. In the second category, we primarily focus on two specific scenarios:

- **Parameter Identification.** Solving the parameter vector based on the information from observation vectors: In this scenario, we aim to estimate the unknown parameters of the system based on the available observation vectors. The observation vectors contain information about the system's behavior or response under conditions that are characterized by the parameter vector.

- **State Estimation.** Solving the field distribution based on the information from observation vectors: In this scenario, the objective is to determine the unknown field distribution of the nuclear system based on the available observation vectors.

F. Contribution of this Work

A previous work [13, 15, 16] demonstrated the feasibility of employing the RODT framework in engineering applications. However, a need exists for a systematic and in-depth investigation of the core components of RODT, namely parameter identification and state estimation. In addition, the development of mature and modular software is crucial to facilitating the application of RODT in practical scenarios. This motivated us to develop an efficient and appropriate surrogate inverse model to accurately predict the parameters and states. This also motivated our research on hybrid optimization approaches that can effectively balance coarse- and fine-grid search

(coarse–fine-grid search) methods, which are discussed in the following sections.

The contributions of this research are as follows:

- Investigation and implementation of novel gradient-free optimization algorithms specifically tailored to handle the challenges posed by the discontinuous surrogate forward mapping constructed with the KNN forward model of our RODT and comparing the convergence profile, stability, and accuracy-time performance.

- Exploration and evaluation of the proposed coarse–fine-grid search methods within various global optimization approaches, aiming to strike a balance between accuracy and computational efficiency in parameter identification and state estimation.

- Development and deployment of efficient modular RODT software integrated with the aforementioned algorithms into the inverse problem solver, which applies to other sustainable energy systems.

These contributions focus on advancing state-of-the-art RODT by addressing the unique challenges encountered in solving inverse problems in nuclear engineering and proposing novel methodologies for improved parameter identification and state estimation. Furthermore, our work addresses the critical challenge of achieving real-time, high-precision simulations in nuclear energy DT applications, significantly enhancing the capabilities of DTs in the nuclear energy sector and facilitating improved monitoring, control, and optimization of nuclear energy systems.

II. METHODOLOGY

The modeling process of an RODT is classified into offline and online stages.

- **Offline phase**: training a non-intrusive ROM, which includes (i) preparation of the full order snapshots; (ii) preparation of the reduced basis; (iii) preparation of the coefficient; and (iv) preparation of the surrogate forward model to predict the coefficient with the information of the parameter.

- **Online phase**: parameter identification and state estimation, which includes (i) building an inverse model to predict the parameter based on the information of the observation data (clean or with observation noise) and (ii) building an inverse model to predict the coefficient based on the information of the observation data (clean or with observation noise).

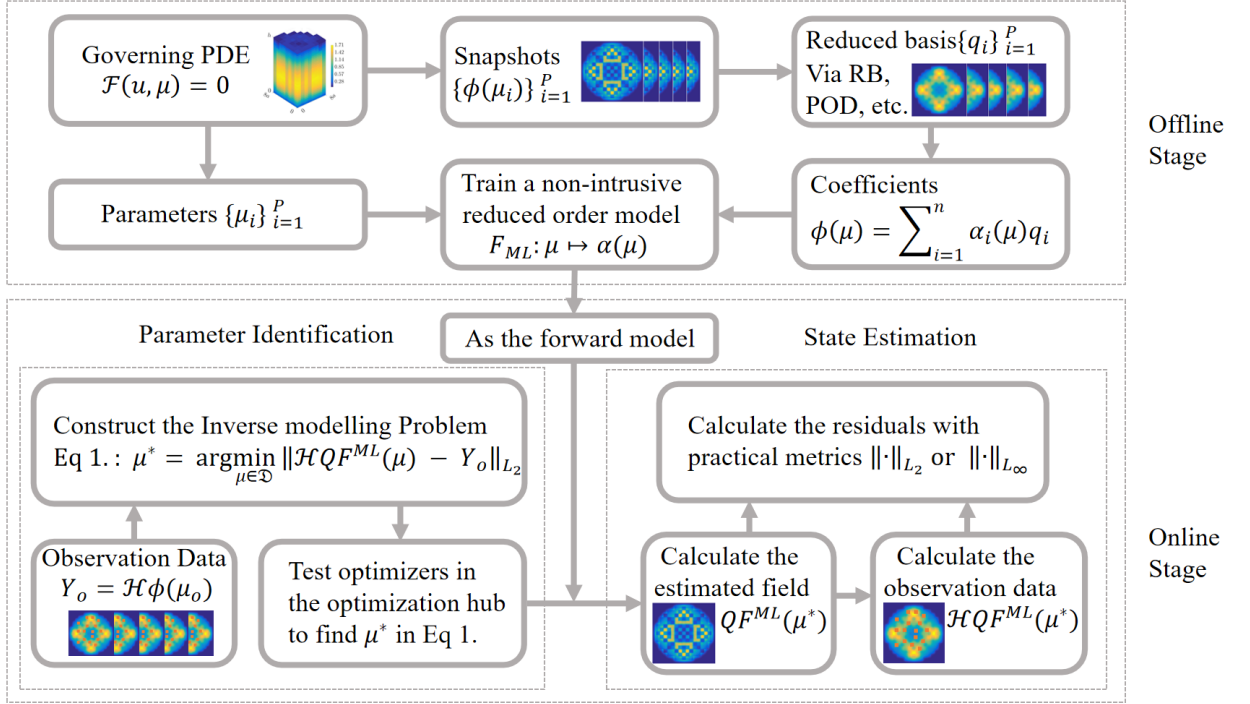The methodology is illustrated in the following flowchart.

3

Fig. 1: Working flow of the RODT schema.

A. Offline phase: Training a Non-Intrusive ROM

The variation in the power field Φ in a nuclear core is characterized by physical laws that are generally described implicitly by a governing equation [52], such as the neutron transport or diffusion equations, which is written as Equation 2.1.

$$\begin{cases} \mathcal{F}(\Phi(r, \boldsymbol{\mu}), \boldsymbol{\mu}) = 0 \\ r \in \Omega \subset \mathbb{R}^d \\ \boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p \end{cases}, \qquad (2.1)$$

where $\Omega \subset \mathbb{R}^d$ represents the spatial domain of dimension d, with $d \geq 1$, and $\phi(\boldsymbol{\mu}) = \Phi(r, \boldsymbol{\mu})$ is defined in a Hilbert space that is equipped with an inner product $\langle \cdot, \cdot \rangle$, and the induced norm $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$. $\mathcal{D} \subset \mathbb{R}^p$ represents the p-dimensional feasible parameter domain that covers the operation of the reactor core. In this study, Eq. 2.1 represents two-group neutron diffusion equations [53].

1. Preparation of the Full-Order Snapshots

In this study, the numerical solution of Eq. 2.1 was obtained using the proprietary CORCA-3D code package developed at the Nuclear Power Institute of China (NPIC) [54]. The CORCA-3D code is a black-box solver; it can be substituted with other proprietary codes as needed.

To construct a non-intrusive ROM, it is necessary to gather a set of full-order solution snapshots. We sample $\boldsymbol{\mu}_i$ in the parameter domain $\mathcal{D}$ to obtain the discrete set $\boldsymbol{D} = \{\boldsymbol{\mu}_i \in \mathbb{R}^p \mid \boldsymbol{\mu}_i \in \mathcal{D}, i = 1, \ldots, P\}$, representing the entire set $\mathcal{D}$. Using CORCA-3D, we obtain a solution snapshot set $\boldsymbol{M} = \{\phi(\boldsymbol{\mu}) \in \mathbb{R}^N \mid \boldsymbol{\mu} \in \boldsymbol{D}\}$.

The knowledge of this manifold is implicitly dependent on the knowledge of the parameters $\boldsymbol{\mu}$, and for better construction of a non-intrusive model, we endeavor to find a low-rank representation for the solution snapshot set $\boldsymbol{M}$. Reduced-basis methods (RB) [55–57] indicates that in state where Φ is sufficiently regular, it can be approximated by an n-dimensional reduced basis $\{q_i\}$, as follows:

$$\phi(\boldsymbol{\mu}) \cong \phi_n(\boldsymbol{\mu}) = \sum_{i=1}^{n} \alpha_i(\boldsymbol{\mu}) q_i. \qquad (2.2)$$

For simplicity, we denote the approximation of the solution manifold as follows:

$$\phi_n(\boldsymbol{\mu}) = \boldsymbol{Q}\alpha(\boldsymbol{\mu}), \qquad (2.3)$$

where $\boldsymbol{Q} \in \mathbb{R}^{N \times n}$ represents the assembled basis $N \times n$-dimensional matrix $\{q_i, i = 1, ..., n\}$, and $\alpha(\boldsymbol{\mu}) \in \mathbb{R}^n$ denotes the assembled coefficient n-dimensional vectors $\{\alpha_i, i = 1, ..., n\}$.

2. Preparation of the Reduced Basis

For simplicity and time efficiency, we chose the singular value decomposition (SVD) method for generating the reduced basis. The steps are as follows:

Correlation Matrix. The correlation matrix $\mathbf{C}$ is computed by taking the inner product of each pair of snapshots, as expressed by the following equation:

$$\mathbf{C}_{i,j} = \frac{1}{P}\langle \phi(\boldsymbol{\mu_i}), \phi(\boldsymbol{\mu_j})\rangle, \quad \forall 1 \le i, j \le P. \quad (2.4)$$

Here, $\langle \phi(\boldsymbol{\mu_i}), \phi(\boldsymbol{\mu_j})\rangle$ represent the inner product of the ith and jth snapshots. Subsequently, the eigenvalues λ_i and corresponding eigenvectors v^i of the correlation matrix $\mathbf{C}$ are computed.

Proper orthogonal decomposition (POD). The jth POD basis vector q^j, which is independent of parameter $\boldsymbol{\mu}$, is obtained as a linear combination of snapshots.

$$q^j = \sum_{i=1}^{N} v_i^j \phi^i. \quad (2.5)$$

In this equation, v_i^j represents the ith element of the j-th eigenvector, and ϕ^i corresponds to the i-th snapshot. The magnitude of the j-th eigenvalue λ_j provides information about the relative importance of the jth POD basis vector.

Collection of the best basis minimizing the Frobenious error. The solution snapshot set is denoted as $M = \{\phi(\boldsymbol{\mu_k})\}_{k=1}^{P} \subset \mathbb{R}^N$. When applied to computation, we define the snapshot matrix as $\boldsymbol{S} \in \mathbb{R}^{N \times P}$, which contains snapshots $\phi(\boldsymbol{\mu_k})$ as columns. The first n POD bases are assembled into a matrix $\boldsymbol{Q}_n = [q_1, \dots, q_n] \in \mathbb{R}^{N \times n}$. Among all orthonormal bases of size n, the POD basis minimizes the Frobenious norm least squares error (defined in the form of $\|\bullet\|_{\mathrm{F}} = \sqrt{\sum_i^m \sum_j^n |\bullet_{ij}|^2}$) of the reconstruction of snapshot matrix $\boldsymbol{S}$, which further corresponds to the Echart–Young theorem [58]:

$$\min_{\boldsymbol{Q}_n} \left\| \boldsymbol{S} - \boldsymbol{Q}_n \boldsymbol{Q}_n^T \boldsymbol{S} \right\|_F^2 = \sum_{k=n+1}^{N} \lambda_k, \quad (2.6)$$

where λ_k represents the kth singular value of matrix S. In summary, the POD basis constitutes a set of orthonormal vectors that offers the most effective n-dimensional representation of the given snapshots.

3. Preparation of the Coefficient

Following the aforementioned procedure, the coefficient is constructed as the projection of the snapshot matrix S onto the vector space spanned by the POD basis Span$\{q_1, q_2, \dots, q_n\}$

For any $\phi(\boldsymbol{\mu})$ belonging to the set $\mathcal{S}$, the n-dimensional approximation of $\phi(\boldsymbol{\mu})$ is given by Equation 2.2. In this manner, the coefficient can be computed using an orthogonal projection as follows:

$$\alpha(\boldsymbol{\mu}) = \sum_{i=1}^{n} \frac{\langle \phi(\boldsymbol{\mu}), q_i \rangle}{\langle q_i, q_i \rangle} = \sum_{i=1}^{n} q_i^{\mathrm{T}} \phi(\boldsymbol{\mu}), \quad (2.7)$$

where $\langle q_i, q_i \rangle$ equals 1 owing to the orthogonormality of the POD basis.

Moreover, in matrix form, the coefficient matrix $\boldsymbol{A}_n = [\alpha_1(\boldsymbol{\mu}), ..., \alpha_n(\boldsymbol{\mu})]^{\mathrm{T}} \in \mathbb{R}^{n \times P}$ follows the following relationship:

$$\boldsymbol{S} \cong \boldsymbol{S}_n = \boldsymbol{Q}_n \boldsymbol{A}_n = \boldsymbol{Q}_n \boldsymbol{Q}_n^T \boldsymbol{S}. \quad (2.8)$$

4. Training a Forward Model

The goal of the forward model is to construct an efficient surrogate mapping model between the parameter and coefficient, as follows:

$$\alpha(\boldsymbol{\mu}) \cong F^{ML}(\boldsymbol{\mu}), \quad (2.9)$$

where F^{ML} denotes the surrogate forward mapping constructed by the machine learning approach, such as model-dependent machine learning methods or model-free optimization searching methods.

The response of the ROM model, denoted as $F^{ML}(\boldsymbol{\mu})$, corresponds to the predicted coefficient data. The image of space $\boldsymbol{D}$ under the forward map $F^{ML}(\boldsymbol{\mu})$ represents the set of responses for all possible states. The difference $Y_o - \mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu})$ is referred to as the observation data misfit or the residuals associated with parameter $\boldsymbol{\mu}$.

We design the cost function as Equation 2.12 based on the residual. The forward model is used in the search of the optimal approximation for the operator ϕ in the functional set $\{\boldsymbol{Q}F^{ML}, F^{ML} \in \mathcal{E}(\boldsymbol{D}, \mathbb{R}^n)\}$, where $\mathcal{E}(\boldsymbol{D}, \mathbb{R}^n)$ denotes the set of functions mapping from $\boldsymbol{D}$ to $\mathbb{R}^n$, with n the reduced dimension in the MOR process.

Previous research [13] evaluated various forward model approaches, among which the KNN algorithm demonstrated exceptional accuracy and efficiency. In this study, we specifically selected the KNN algorithm [48] as the forward model to exploit its full potential. To enhance the performance of the KNN algorithm, we focused on two crucial parameters that significantly influence its efficacy.

The first parameter relates to the voting distance in KNN, wherein we investigate the impact of utilizing either the Euclidean distance or Manhattan distance as the metric for determining the nearest parameter choice in the training set. Notably, these two distances can be expressed as Minkowski norms with an exponent of

$p = 1, 2$, corresponding to the Euclidean and Manhattan distances, respectively. The Minkowski distance, a measure of the discrepancy between the testing and training parameters, can be mathematically represented as follows:

$$\left\| \boldsymbol{\mu}^{test} - \boldsymbol{\mu}^{train} \right\|_p = \left(\sum_{i=1}^{dim(\boldsymbol{\mu})} |\boldsymbol{\mu}_i^{test} - \boldsymbol{\mu}_i^{train}|^p \right)^{\frac{1}{p}},$$

(2.10)

where $dim(\boldsymbol{\mu})$ denotes the dimension of the parameter $\boldsymbol{\mu}$. Finally, we choose the Euclidean metric.

Another vital parameter is K, which refers to the number of candidates voted for. The optimal choice of K can be determined by plotting the relative L_2 error of the predicted observation and true vectors, which is expressed as follows.

From Figure 2(a), we can deduce that when K = 5, our forward model using KNN performs with the best prediction accuracy, and the variation in the error is relatively smooth, which corresponds to the continuous variation in the value of the field at the corresponding position of the field manifold snapshot.

In contrast, the stair-shaped observation data predictions in Figure 3 demonstrate that the KNN forward surrogate model has a step-like progression as neighbors are adjusted between discrete datasets and thus are not continuous or differentiable. In this manner, well-developed state-of-the-art algorithms based on gradient information for solving optimization and inverse problems did not work well in our case. To meet this demand, we investigated other novel approaches, which are discussed in detail in the following sections.

B. Online phase: Parameter Identification and State Estimation

The inverse problem in nuclear engineering refers to the task of determining the model parameters that yield the observed data, as opposed to the forward problem, which involves predicting the data based on the given model parameters. The objective is to determine the model parameters $\boldsymbol{\mu}^*$ that approximately satisfy Equation 2.12.

The nature of the inverse problem depends on whether operator ϕ is linear or nonlinear. In most cases, particularly when solving systems governed by the neutron flux diffusion function, the inverse problem is nonlinear, owing to the nonlinearity of the forward map. Thus, we attempt to develop nonlinear algorithms, gradient-free algorithms, or algorithms based on the initial guess given by fast linear algorithms, such as KNN.
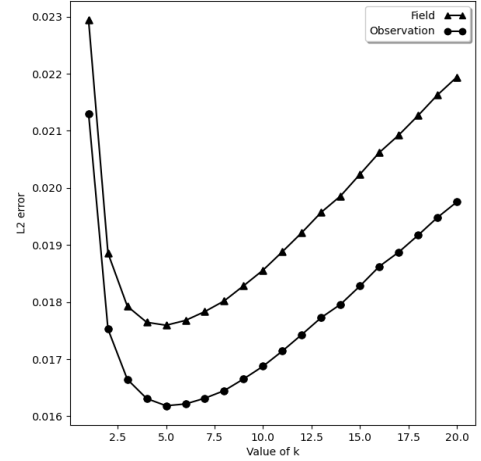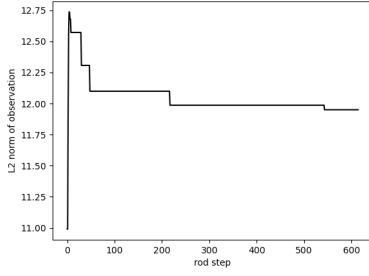

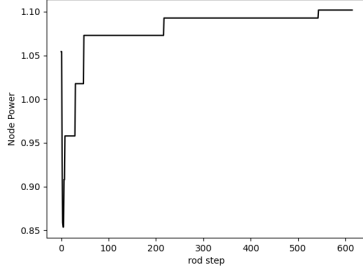((a)) KNN Forward model
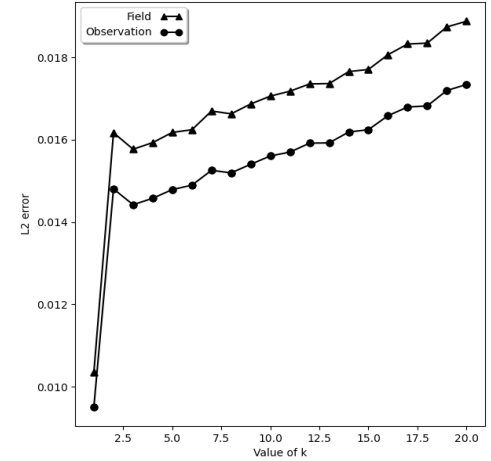
((b)) KNN Inverse model

Fig. 2: Finding the optimal K for the KNN model (Forward on the left, Inverse on the right).

#### 1. Setup of Inverse Model and Optimization

In the context of optimal control theory, the governing equations that describe the behavior of a physical system are commonly referred to as state equations. However, in many practical scenarios, interest lies not only in the physical state itself but also in its impact on certain objects or quantities. Furthermore, only a limited amount of data can often be obtained from the physical state. To address these considerations, an additional operator denoted by $\mathcal{H}$, known as the observation operator, is introduced. This operator maps the state of the physical system denoted by Φ to the desired observations denoted

((a)) L_2 norm of predicted observation

((b)) 1th vertical level, the 43th assembly
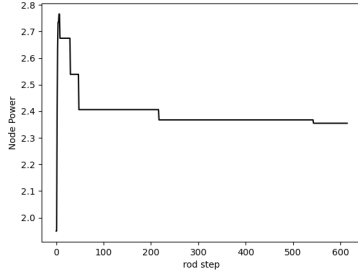
((c)) 18th vertical level, the 11th assembly

Fig. 3: Non-differentiable stair-shaped observation data prediction via the KNN surrogate forward model (tested when the first parameter S_t varies from 0 to 615 steps, the other parameters take the midpoint values of the parameter intervals, respectively, 1250 for B_u; 50 for P_w; and 295 for T_{in}).

by $Y_o \in \mathbb{R}^m$, where m denotes the total number of observations. Given a field Φ, the observation vector Y_o is given by the following equation and approximated by an error:

$$Y_o = \mathcal{H}\Phi = \mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu}) + e. \qquad (2.11)$$

where $e \in \mathbb{R}^m$ represents the observation noise vector that captures the presence of observational errors. The quality of a given set of model parameters $\boldsymbol{\mu}$ is evaluated using the L_2 distance between the simulated and practical observations, which is quantified using the following

cost function:

$$J(\boldsymbol{\mu}) := \left\| \mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu}) - Y_o \right\|_{L_2}^2. \qquad (2.12)$$

The objective of the inverse problem is to minimize the observation error, which can be formulated as

$$\boldsymbol{\mu}^* = \operatorname*{argmin}_{\boldsymbol{\mu} \in \mathcal{D}} J(\boldsymbol{\mu}) := \operatorname*{argmin}_{\boldsymbol{\mu} \in \mathcal{D}} \left(\left\| \mathcal{H}\boldsymbol{Q}F^{\mathrm{ML}}(\boldsymbol{\mu}) - Y_o \right\|_{L_2}^2 \right). \qquad (2.13)$$

It is important to note that the cost function described above is possibly nonlinear and non-differentiable with respect to the discontinuous function F^{ML}. Therefore, we developed the following algorithms to solve the optimization problem:

III. COMPARISON OF DIFFERENT APPROACHES FOR SOLVING INVERSE MODELS

To solve inverse problems, machine learning and optimization methods can be employed to search for the optimal parameters or field distributions that best fit the observed data [59]. Additionally, as mentioned in Section II A 4, when we employ the KNN algorithm to construct the surrogate forward model, the surrogate forward mapping is non-differentiable and discontinuous. This poses a challenge because the traditional approaches used to solve continuous optimization and inverse problems [60] are not applicable in this context. This necessitates the exploration of novel inverse problem-solving methods suitable for our particular problem. Hence, this study investigates various deterministic optimization algorithms, metaheuristic algorithms, and their hybrids to address this challenge.

A. Exhaustive Direct Search with Latin Hypercube Sampling (LHS)

To evaluate and compare the different approaches to solving inverse models, it is essential to establish a benchmark for measuring their performance. In this section, we propose using exhaustive direct search (EDS) with latin hypercube sampling (LHS) as the benchmark method for inverse problems. It combines the accuracy of EDS methods, which have been widely used since the 1960s [61], with the sampling efficiency of LHS.

LHS [62] is a sampling technique that ensures a comprehensive exploration of the solution space while maintaining desirable properties. The range of each variable was divided into equally probable intervals, and the sample points were strategically placed to satisfy the requirements of a Latin hypercube, where each sample was the only one in its corresponding axis-aligned hyperplane. The advantage of LHS lies in its independence from the

number of dimensions, which allows for efficient sampling without the need for an increasing number of samples. Furthermore, LHS facilitates sequential sampling, enabling the tracking of the already selected samples.

B. Exhaustive Direct Search with 2 Steps Latin Hypercube Sampling (LHS2STEPS)

To enhance the benchmark algorithm and mitigate the uncertainties introduced by the randomness of the sampling process, [13] introduced a two-step sampling strategy. This strategy involves further sampling the neighborhoods of the n_1 best candidates generated during the initial stages of the LHS algorithm. For the detailed pseudocode, please refer to 1, excluding the part related to ALHS.

C. Exhaustive Direct Search with Assembled 2 Steps Latin Hypercube Sampling (ALHS)

To improve our benchmark algorithm and further reduce the uncertainty caused by the randomness of the sampling process, we introduce a mean mechanism at the end of the classical LHS algorithm to calculate the mean of the n_1 top-rated candidates in the five clones, which are spread around the n_2 top-rated candidates in the first sampling process. The concept is illustrated in Algorithm 1.

D. Efficient Deterministic Machine Learning Algorithm: KNN

In addition to the benchmark LHS method, another efficient deterministic algorithm that can be considered for solving inverse models is the KNN algorithm. KNN is a nonparametric classification and regression algorithm that can be adapted to solve inverse problems.

The KNN algorithm operates based on the principle of similarity. Given a new input data point, KNN finds the K-nearest neighbors in the training dataset based on the Euclidean distance and then generates the output value by averaging the output values of the K-nearest neighbors. The selection of the optimal value of K can be determined by graphing the relative L_2 error between the predicted observation vector and the true vector, as shown in Figure 2(b). The analysis in Figure 2(b) reveals that selecting K = 1 yields the optimal outcome in terms of minimizing the L_2 prediction error, thereby emphasizing its advantageous accuracy.

E. Metaheuristics Algorithms with Physical Constraints

Recently, the application of metaheuristic algorithms with physical constraints [63–65] has gained significant attention for inverse modeling. These algorithms offer powerful gradient-free optimization techniques that can effectively handle complex inverse problems while satisfying the physical constraints imposed by system dynamics.

In this manner, we implemented and adapted practical algorithms and constructed related solvers that were integrated into the developed RODT framework. These solvers are designed to address the inverse problem presented by Equation 2.12.

1. Differential Evolution Algorithm (DE)

The differential evolution (DE) algorithm [66] offers a promising approach for tackling inverse modeling problems. By leveraging its ability to handle nonlinear and nonconvex objective functions, DE can effectively explore the parameter space. Furthermore, DE provides mechanisms to incorporate physical constraints, ensuring that the generated solutions adhere to system dynamics. The concrete procedure of the DE algorithm can be found in a previous work [13], in which we used the best mutation strategy recommended in [67] for the mutation step in DE, and the mutation donor $V_{i,G}$ is given by Equation 3.14.

$$V_{i,G} = \mu_{r_5,G} + F \cdot (\mu_{r_1,G} - \mu_{r_2,G} + \mu_{r_3,G} - \mu_{r_4,G}), \tag{3.14}$$

where $r_{k,G}$, $k \in \{1, 2, 3, 4, 5\}$ are randomly chosen from $\{1, 2, ...NP\}$, NP is the number of populations, μ is a parameter, and F and G denote the scaling factor and the number of generations, respectively.

2. Particle Swarm Optimization Algorithm (PSO)

Particle swarm optimization (PSO) [68] is a population-based optimization technique inspired by the collective behavior of bird flocking or fish schooling. It involves particles representing potential solutions that update their positions based on the personal and global best positions. The algorithm iterates a set number of times by adjusting the particle velocities and positions. The best solution, represented by the global best position with the lowest objective function value, is returned.

Algorithm 1 Exhaustive Direct Search with Assembled 2 Steps Latin Hypercube Sampling
---

Inputs:
Observations: Y_o
Initial guess: $\boldsymbol{\mu}_{\text{initial}}$
Forward function: F^{ML}
Transformation operator: H
First stage sampling number: $n^{s,1}$
Second stage sampling number: $n^{s,2}$
$\mathcal{U}^1 = \text{LHS} \left(\text{ number } = n^{s,1}, \text{ initial } = \boldsymbol{\mu}_{\text{initial}} \right)$
$\boldsymbol{\mu}_1^* = \underset{k=1..n^s}{\operatorname{argmin}} \left(\left\| HV_n F^{\text{ML}} \left(\boldsymbol{\mu}^k \right) - Y_\circ \right\|_2 \right)$
for i from 2 to n_1 **do**   ▷ $n_1 = 5$ in this article.
     $\mathcal{U}^1 = \text{LHS} \left(\text{ number } = n^{s,1}, \text{ initial } = \boldsymbol{\mu}_{\text{initial}} \right)$
     $\boldsymbol{\mu}_1^* = \underset{k=1..n^s}{\operatorname{argmin}} \left(\left\| HV_n F^{\text{ML}} \left(\boldsymbol{\mu}^k \right) - Y_\circ \right\|_2 \right)$
end for
for j from 1 to n_1 **do**
     $\mathcal{U}_j^2 = \text{LHS} \left(\text{ number } = n^{s,2} \text{ the initial } = \boldsymbol{\mu}_i^* \right)$
end for
for k from 1 to n_2 **do**   ▷ Start of the main part of ALHS, $n_2 = 10$ in this article.
     $\boldsymbol{\mu}_k^* = \underset{\boldsymbol{\mu} \in \bigcup \mathcal{U}_j^2}{\operatorname{argmin}} \left(\left\| HV_n F^{\text{ML}}(\boldsymbol{\mu}) - Y_\circ \right\|_2 \right)$
end for   ▷   End of the main part of ALHS.
$\overline{\boldsymbol{\mu}} = \left(\sum_{k=1}^{10} \boldsymbol{\mu}_k^* \right) / 10$
Outputs:
Parameter: $\overline{\boldsymbol{\mu}}$
Reconstructed field: $V_n F^{\text{ML}} \left(\boldsymbol{\mu}^* \right)$

---

3. Fast Simulated Annealing Algorithm (FSA)

The simulated annealing (SA) [69] algorithm, which was initially proposed for solving the well-known travelling salesman problem (TSP) [70], is a popular approach for solving optimization problems in various fields [71], including nuclear engineering [72].

SA was inspired by metallurgical annealing, in which a metal is heated and gradually cooled to achieve a more ordered configuration. Similarly, the SA starts from a high-energy state (initial solution) and progressively lowers the temperature until it converges to a state of minimal energy (optimal solution).

The fast simulated annealing (FSA) [73] is an adaptation of classical simulated annealing (CSA) [69], which improves the computational efficiency and convergence speed. CSA utilizes a local sampling approach and controls the fluctuation variance through an artificial cooling temperature, denoted as $T_c(t)$. FSA modifies the cooling schedule to accelerate convergence using a cooling temperature $T_f(t)$ that decreases reciprocally rather than logarithmically with time t.

4. Cuckoo Search Algorithm (CS)

In the cuckoo search (CS) algorithm [74], nests are used to represent potential solutions, and eggs within the nests symbolize candidate solutions. Through an iterative process, the algorithm generates new solutions by employing Lévy flights and subsequently evaluates their fitness. If a newly generated solution is superior to an existing one within a given nest, it replaces the incumbent solution. To maintain population diversity and improve the overall quality of solutions, the population is sorted based on fitness, and a predetermined number of solutions are replaced with new solutions generated via random walks. The algorithm terminates upon reaching a predefined maximum number of generations, and the best solution, characterized by the lowest objective function value, is returned as the final outcome.

CS utilizes a balanced combination of a local random walk and a global explorative random walk, controlled by a switching parameter p_a. A local random walk is defined as follows:

$$\mu_i^{t+1} = \mu_i^t + \alpha \otimes H \left(p_a - \epsilon \right) \otimes \left(\mu_j^t - \mu_k^t \right). \quad (3.15)$$

In the above equation, μ_j^t and μ_k^t represent two solutions randomly selected through permutations. $H(\bullet)$ denotes the Heaviside function, $\otimes$ represents point-to-point multiplication, ϵ is a random number drawn from a uniform distribution, and s represents the step size. The global random walk, on the other hand, is conducted using Lévy flights, as follows:

$$\mu_i^{t+1} = \mu_i^t + \alpha L(s, \lambda), \quad (3.16)$$

where $L(s, \lambda)$ is defined as

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0).$$
(3.17)

5. Artificial Neural Network (NN)

The neural network (NN) algorithm [75], inspired by the behavior of the human brain, simulates interconnected artificial neurons to process information, learn from data, and estimate unknown parameters or states. They consist of interconnected artificial neurons organized into layers that receive input signals, process them through weighted connections, and produce output signals using activation functions. NNs can learn from input-output examples by adjusting the connection weights during training, enabling them to capture underlying patterns and correlations, making them suited for solving physical problems [76].

The combination of the Adam and L-BFGS optimizers in an alternating manner has demonstrated efficacy in training NNs for complex problem solutions [77]. This approach leverages the strengths of the two optimizers to enhance training and improve the overall network performance.

However, owing to NN's continuous property, it cannot handle discrete problems well, particularly when the test data are noisy, which is further discussed in Section V.

F. Hybrids of Optimization Methods for Inverse Problem: Balancing Coarse–Fine-Grid Search

To address the challenges posed by the discontinuity of KNN surrogate forward mapping and noisy observations, we propose hybrid algorithms that combine the strengths of deterministic machine-learning methods for coarse-grid search and metaheuristic algorithms for fine-grid search. This methodology can also be found in fields that encompass feature extraction of aerial data [78], automatic free-form optics design [79], and hyperparameter optimization [80]. The objective of a coarse–fine-grid search is to strike a balance between global exploration and local exploitation [81] to achieve accurate, efficient, and less sensitive parameter estimation. The hybrid optimization hub designed accordingly can be further illustrated by the following schema:

1. Cuckoo Search with Differential Evolution (CSDE)

In the later stages of the standard CS process, notable issues arise in the form of information waste and slow convergence. These issues stem from the execution of independent evaluations and the lack of sufficient mechanisms for information sharing within the population. Consequently, valuable information fails to propagate effectively, resulting in suboptimal convergence rates and reduced search efficiency.

From an alternative perspective, the underlying biological nature of mutations within the standard DE algorithm was explored. In this context, a novel mathematical elucidation of these mutations is sought. Notably, researchers in the natural sciences have observed that mutations can be understood as sequences of transformations that transpire within a DNA molecule. Furthermore, the transformations pertaining to modifications exhibited a statistical distribution that adhered to the Levy distribution. This finding suggests an analogy between mutations and Levy flight [82]. Drawing inspiration from this observation, it is conceivable to integrate the Levy flight process into the generation of the DE population.

In this regard, we developed the cuckoo search differential evolution (CSDE) algorithm, which lies within the framework of the CS algorithm, integrating the DE mutation operation into the production of the population for each generation. The first step of CSDE is the initialization: Define the objective function $f(\bullet)$. We set the population size N, problem dimension D, discovery probability Pa, step size α, scaling factor F, crossover probability CR, maximum number of iterations T, and search domain range $[\mu_{lb}, \mu_{ub}]$. The steps are presented in Algorithm 2.

2. Advanced CSDE (ACSDE) and Advanced DE (ADE)

We utilize the SVC algorithm to classify the test parameters and denote the center of the classified hypercube as the initial machine learning estimate μ_{SVC}; then, we use the CSDE and DE algorithms to further explore the parameter space with the initial population spread around the center μ_{SVC}. In the SVC process, according to a previous research [15] on the importance rate of the input parameters, the last parameter, temperature Tin, contributes slightly, and we also consider that Tin varies at a relatively small boundary; thus, we only administer the SVC process to the first three parameters. We chose two representative metrics to evaluate the performance of the SVC model.

Confusion Matrix. The confusion matrix [83] provides a comprehensive overview of the binary classification results by displaying the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. This allowed us to assess the accuracy of the model' by differentiating classes and identifying potential sources of misclassification. We can infer from Figure 5 that our SVC classification model predicts the first two parameters with relatively high accuracy while exhibiting confusion concerning the third
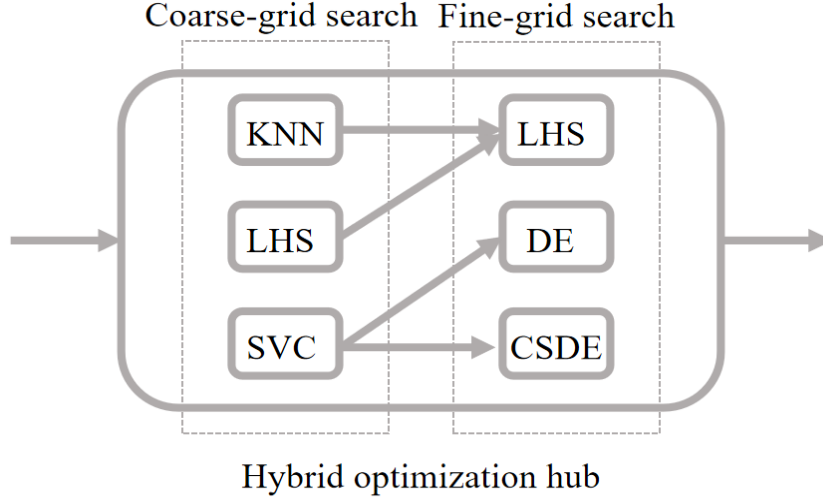
Fig. 4: Hybrid optimization schema of coarse–fine-grid search.

parameter.

F1-score. The F1-score [84], defined in Equation 3.18, is a popular performance metric for binary classifier models. In Equation 3.18, TP, FN, FP are the numbers of true positives, false negatives, and false positives classified by the model, respectively.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \qquad (3.18)$$

The predictive model employed in this study for each parameter is a multi-class SVC utilizing the "one-vs-others" strategy. This approach involves training separate binary classifiers to distinguish between the candidate intervals for a given parameter. Consequently, performance evaluation necessitates the adoption of the Macro F1-score as a metric. The Macro F1-score 3.19 is computed as the average of the F1-scores obtained for each interval, reflecting the overall effectiveness of the classifiers in capturing intra-parameter variations:

$$\mathrm{F1}_{\mathrm{macro}} := \frac{1}{N} \sum_{i=0}^{N} \mathrm{F1}_i, \qquad (3.19)$$

where i is the interval index, and N is the number of intervals.

Based on the Confusion Matrix shown in Figure 5 and $\mathrm{F1}_{\mathrm{macro}}$, we designed the grid size of the fine-grid search process as follows:

3. K-Nearest Neighbour plus Exhaustive Direct Search with Latin Hypercube Sampling (KNNLHS)

Integrating KNN with LHS provides a robust methodology to address inverse problems. To enhance the comprehensiveness of our research, we standardized the grid size for a coarse–fine grid search. This standardization is consistent with the specifications listed in Table 1.

In conclusion, the k-nearest neighbour plus exhaustive direct search with latin hypercube sampling (KNNLHS) method is a robust and efficient approach compared to similar strategies. This technique combines the exploratory capabilities of KNN in conducting coarse-grid searches with the efficiency and diversity of LHS in sampling local solution spaces. Consequently, the KNNLHS facilitates precise and efficient parameter estimation, accurate navigation of complex objective functions, and the management of noisy datasets. The effectiveness of this approach and its comparative advantages are discussed in Section V.

IV. APPLICATION TO NUCLEAR REACTOR

A. Setup of Reactor Physics Operation Problems

Setup of background information for HPR1000. The objective of constructing the RODT was to accurately predict the power distribution within the HPR1000 reactor core during operation. The core consists of 177 vertical nuclear fuel assemblies, 44 of which are equipped with self-powered neutron detectors (SPNDs) to measure neutronic activity and power fields. Figure 6 provides a visualization of a horizontal slice of the HPR1000 core and an axial slice of an SPND-equipped assembly, with only one-quarter of the core displayed owing to the symmetry along the x and y axes. The gray fuel assemblies represent those containing control rods, whereas the assemblies marked with D indicate the presence of SPNDs. For more detailed information regarding the HPR1000 reactor and generic neutronic physi-

---

Algorithm 2 Cuckoo Search Algorithm with Differential Evolution

---

Step 1: Population initialization
Initialize the population P and the initial position using a random distribution within the search domain range $[\mu_{lb}, \mu_{ub}]$
while $G \leq G_{max}$ **do**
    for each individual μ_i in population P_G **do**
        Step 2: Population updating
        Generate random indices $r1, r2, r3$ (different from i)
        for each dimension j in problem dimension $dim(\mu)$ **do**
            if random$(0, 1) < CR$ **then**
                $\mu_{i,j}^{G} = \mu_{i,j}^{G} + \alpha \otimes (\mu_{r2,j}^{G} - \mu_{r3,j}^{G}) \otimes L(s, \lambda)$
            else
                $\mu_{i,j}^{G} = \mu_{i,j}^{G}$
            end if
        end for
        Step 3 Mutation Step
        Generate random indices $r3, r4, r5$ (different from i)
        Generate a donor vector $V_i^{G} \in \mathbb{R}^{dim(\mu)}$:
        $V_i^{G} = \mu_{r3}^{G} + F \cdot (\mu_{r4}^{G} - \mu_{r5}^{G})$
        Step 4 Crossover Step
        Generate a trial vector $U_i^{G} \in \mathbb{R}^{dim(\mu)}$:
        for $j = 1$ to $dim(\mu)$ **do**
            if $random(0, 1) \leq Cr$ **then**
                $U_{i,j}^{G} = V_{i,j}^{G}$
            else
                $U_{i,j}^{G} = \mu_{i,j}^{G}$
            end if
        end for
        Step 5 Selection Step
        Evaluate the trial vector U_i^{G}
        if $f(U_i^{G}) < f(\mu_i^{G})$ **then**
            $\mu_i^{G+1} = U_i^{G}$
        else
            $\mu_i^{G+1} = \mu_i^{G}$
        end if
        Step 6 Abandon nests
        Sort the population P_G based on the fitness values in ascending order.
        Determine the number of solutions to be replaced, $n_{\text{replace}} = \text{rand}(p_a \cdot NP)$.
        for $i = 1$ to n_{replace} **do**
            Generate a new solution X in the search space
            with random walk given in Equation 3.15.
            Evaluate the objective function f for X.
            if $f(X) < f(\text{worst solution in} P_G)$ **then**
                Replace the worst solution in P_G with X.
            end if
        end for
    end for
    $G = G + 1$
end while

TABLE 1: Grid size design for the Coarse–Fine-Grid search based on the SVC prediction evaluation.

ParameterNames	S_t	B_u	P_w
F1$_{\mathbf{macro}}$	9.98E-01	9.93E-01	5.18E-01
Size coarse-grid(SVC)	41 steps	250 MWd/tU	8% FP
Size fine-grid(DE/CSDE)	41 steps	250 MWd/tU	20% FP
Relative Size coarse-grid(SVC)	6.67%	10.00%	10.00%
Relative Size fine-grid(DE/CSDE)	6.67%	10.00%	25.00%
Parameter Boundary	[0, 615]	[0, 2500]	[20%, 100%]

((a)) S_t
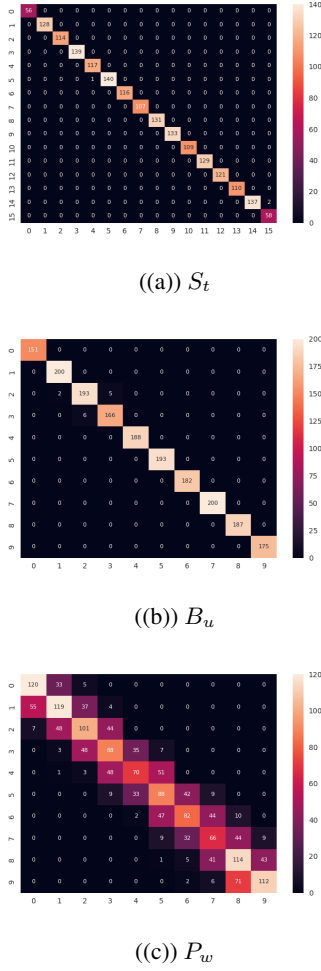

((b)) B_u

((c)) P_w

Fig. 5: Confusion Matrix Plot that evaluates the efficacy of SVC prediction in the first 3 parameters (S_t, B_u, P_w, respectively).

cal model, please refer to [14]. Further descriptions of the model for data assimilation and the initial work on RODT can be found in [35] and [13], respectively.

During the normal operation of the HPR1000 reactor, two types of control rods were utilized for regulation. The first type, known as compensation rods, is responsible for coarse control and a substantial reactivity reduction. These compensation rods comprise four subtypes: G1, G2, N1, and N2. The second type, called regulating rods (R rods), are employed for fine adjustments to maintain the desired power or temperature [85]. Power evolution within the reactor is influenced by various factors, including the movement of the control rods, the burnup of nuclear fuel, variations in the power level of the reactor core, and fluctuations in the inlet coolant temperature. This evolution was mathematically modeled using two-group diffusion equations [13] and numerically solved using the CORCA-3D code package [86]. Devel-

oped at NPIC, the CORCA-3D code is capable of solving 3D few-group diffusion equations, considering thermal-hydraulic feedback, and performing pin-by-pin power reconstruction.

Setup of parameter domain. In this study, we consider CORCA-3D to be an opaque solver in which we input the parameter vector $\boldsymbol{\mu}$ and obtain the output Φ. In particular, the power field Φ is constrained to be dependent on a set of "general" parameters that indicate the stage of the reactor's life cycle:

$$\mu := (S_t, B_u, P_w, T_{in}). \qquad (4.20)$$

- S_t: The control rod insertion steps, ranging from 0 to 615, represent the movement of the compensation rods from all rod clusters being out (ARO) to fully inserted. This range takes into account the overlap steps.

- B_u: The average burnup of the fuel in the entire core indicates the amount of energy extracted from the fuel and increases over time. Its value ranges between 0 (at the beginning of the fuel's life cycle) and $B_{u,\max}$, which is set to 2500 MWd/tU (the end of the fuel's life cycle). The specific evolution of B_u depends on the reactor's operational history.

- P_w: The power level of the reactor core ranges from 0.3 to 1 FP (full power).

- T_{in}: The temperature of the core coolant at the inlet falls within the range of 290 to $300°C$.

In this manner, Φ can be implicitly represented by $\phi(\boldsymbol{\mu}) = \phi(S_t, B_u, P_w, T_{in})$, thanks to CORCA-3D. There are 177 fuel assemblies in HPR1000, and each assembly is numerically represented using 28 vertical levels. Thus, Φ is a vector with dimensions $N = 4956 (= 177 \times 28)$. The discrete solution set $\boldsymbol{M} = \left\{\phi(\boldsymbol{\mu}) \in \mathbb{R}^N \mid \boldsymbol{\mu} \in \boldsymbol{D}\right\}$ consists of $P = 18480$ solution snapshots with the parameter configuration $D := B_u^s \otimes S_t^s \otimes P_w^s \otimes T_{in}^s$, where $S_t^s = \{0, 1, \ldots, 615\}$, $B_u^s = \{0, 50, 100, 150, 200, 500, 1000, 1500, 2000, 2500\}$, $P_w^s = RU3(30, 100)$, and $T_{in}^s = RU3(290, 300)$. The operator $RU3(a, b)$ represents three independent and identically uniformly distributed samplings in the closed set $[a, b]$ where $a < b$. In this study, we selected 90% of 18480 snapshots for training the forward and inverse models, and the remaining 10% of snapshots were used for test purposes.

Setup of observation data and modeling the observation noise. In the context of RODT, the observations denoted by Y_o are used to infer the input parameter vector $\boldsymbol{\mu}$ and subsequently determine the resulting power field Φ. The arrangement of these observations, organized node by node, is shown in Figure 6. Each observation value represents the average value over the corresponding node in which the SPNDs are located. It is important to clarify that the observations used in the analysis
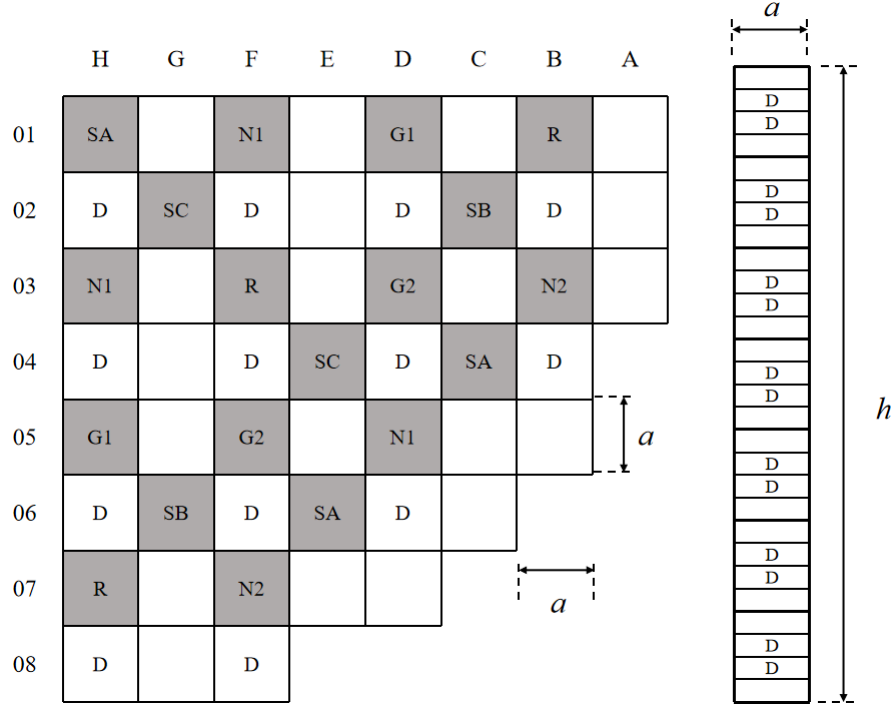
	H	G	F	E	D	C	B	A
01	SA		N1		G1		R	
02	D	SC	D		D	SB	D	
03	N1		R		G2		N2	
04	D		D	SC	D	SA	D	
05	G1		G2		N1			
06	D	SB	D	SA	D			
07	R		N2					
08	D		D					

Fig. 6: Quarter of the core in the radial direction (white square: fuel assembly with SPNDs, grey square: fuel assembly with control rods, D: fuel assembly with neutron detectors) [13].

were not obtained from direct engineering measurements but were derived from numerical simulations of Φ using CORCA-3D.

The noise associated with the observations can be modeled as either a Gaussian or a uniform distribution, depending on the specific requirements. Specifically, the noise term can be mathematically represented as follows:

$$e = Y_o \cdot N(0, \sigma, Y_o) \text{ or } Y_o \cdot U(-\delta, \delta, Y_o)$$

In this context, the vector $N(0, \sigma, Y_o)$ denotes a collection of random variables following a normal distribution. These random variables have the same dimensions as Y_o and are characterized by a mean of zero and standard deviation of σ. Similarly, the vector $U(-\delta, \delta, Y_o)$ represents a set of random variables uniformly distributed within the interval $[-\delta, \delta]$. These variables have the same dimensions as Y_o.

Considering the prevalence of Gaussian noise in real-world nuclear reactors, our subsequent analysis primarily focused on evaluating the performance of the algorithm under Gaussian noise conditions.

Software implementation We integrated our software into the RODT platform using the following modules:

- rodtPro: This module implements data cleaning and preprocessing techniques based on domain expertise.

- rodtROM: This module constructs ROM through techniques like POD, SVD, and RB.

- rodtML: This module focuses on surrogate modeling and inverse model optimization. It also includes hyperparameter optimization techniques.

- rodtPost: This is a post-evaluation module that defines convergence metrics and conducts comparative analyses using evaluation methodologies derived from peer-reviewed literature.

- rodtUI: This module is responsible for visualizing the results, providing an intrinsic review of the methods, showcasing the optimized hyperparameters, and designing user-specific visualizations.

B. Practice Metrics for Inverse Model Evaluation

To evaluate the performance of the algorithms, we designed a series of numerical experiments using nuclear reactor data produced by an intrusive program with added noise. We carefully selected metrics related to accuracy and finally defined the normalized reconstruction fields L_2 error (E_2) and L_∞ error (E_∞) to evaluate the performance of the comparative algorithms.

The reconstructed field L_2 error is defined in Equation 4.21 to evaluate the average estimation of the field, where

$\| \bullet \|_2 := \left(\sum_i \left(\bullet_i \right)^2 \right)^{1/2}$ denotes the L_2 norm.

$$E_2 := \left\| \boldsymbol{Q} F^{\mathrm{ML}} \left(\boldsymbol{\mu}^* \right) - \Phi_{\mathrm{true}} \right\|_2 / \left\| \Phi_{\mathrm{true}} \right\|_2 \quad (4.21)$$

The reconstructed field L_∞ error is defined in Equation 4.22 to evaluate the worst field estimation, where $\| \bullet \|_\infty := \max_i |\bullet_i|$ denotes the L_∞ norm.

$$E_\infty := \left\| \boldsymbol{Q} F^{\mathrm{ML}} \left(\boldsymbol{\mu}^* \right) - \Phi_{\mathrm{true}} \right\|_\infty / \left\| \Phi_{\mathrm{true}} \right\|_\infty \quad (4.22)$$

It should also be noted that the reconstructed observation L_2 error and L_∞ can be generated similarly.

V. NUMERICAL RESULTS

In this section, we describe the experiments conducted to evaluate the performance of parameter identification and state estimation for the proposed algorithms. The evaluation was performed on a test dataset consisting of 40 specimens from HPR1000, as described in Section IV. All the proposed inverse modeling approaches are utilized in the experiments, and it is important to mention that the cost function employed remains consistent with Equation 2.12. Notably, we focused solely on testing the CSDE method and did not include the CS method. Furthermore, we introduce the proposed accuracy metric, illustrated in Section IV B, to assess its performance.

A. Accuracy and Stability Analysis of the Parameter Identification Phase

In this section, we describe the tests conducted to evaluate the accuracy of parameter identification. Initially, we assessed the effectiveness of the proposed hybrid methodology, the coarse–fine–grid search, as discussed in Section III F, using the KNNLHS method as an illustrative example. To visualize this process, we generated scatter plots depicting the predicted values of the parameters S_t and P_w, which were identified as the two dominant parameters in a previous study [15] (see Figure 7 for six cases with different input parameter settings).

Figure 7 shows a visualization of the local search space in the inverse algorithms. The figures in the first row depict scenarios with varying P_w values ranging from relatively low to moderate to relatively high. Correspondingly, the figures in the second row illustrate cases with varying S_t values, again encompassing relatively low, moderate, and high values. These figures effectively show the considerable improvements achieved using the KNNLHS method, particularly in the simultaneous optimization of both P_w and S_t. The visualizations underscore the effectiveness of the method in yielding enhanced optimization outcomes.

The parameter prediction deviation is demonstrated in Figures 8 over the test set. Owing to space limitations,

we chose only KNN, KNNLHS, and FSA for illustration and tested clean and noisy ($\sigma = 5\%$) observation data. Using the LHS algorithm, the KNN prediction for parameter identification became more accurate, and the prediction deviation for the FSA algorithm was concentrated at relatively low intervals, thus proving its excellent accuracy. In general, the accuracy of burnup, the power level, and the control-rod step are acceptable from an engineering perspective.

B. Comparison of the Convergence Rate among Metaheuristic Algorithms

The convergence rate is a crucial criterion for evaluating algorithms that are integrated with the generation design; thus, in this section, we compare different metaheuristic algorithms and their hybrids. Their convergence performance in 50 iterations (generation) with clean and noisy ($\sigma = 1\%$ and $\sigma = 5\%$) observation data is shown in Figure 9.

When the observation data were clean, the FSA exhibited the fastest convergence rate, followed by the advanced differential evolution (ADE). Moreover, when integrated within an SVC coarse-grid search, both the DE and CSDE algorithms demonstrated the ability to escape the local optima in the early stages while incurring a relatively low-cost function, thereby surpassing their standalone performance.

Among the two population-based algorithms, DE outperformed CSDE. This can be attributed to two factors: (i) the adoption of the best mutation method for DE, where we use the optimal mutation strategy for DE while using the classical strategy; and (ii) the relatively straightforward nature of our problem, wherein the inverse problem lacks multiple optima. Consequently, the CS step in each generation of CSDE may lead to unnecessary exploration of the parameter space that is distant from the optimal solution.

In the presence of noise-contaminated observation data, FSA continues to exhibit superior convergence capabilities. However, its advantage diminishes as the level of Gaussian noise reaches 5%, eventually leading to a decline in advantage. Notably, the ADE and advanced cuckoo search differential evolution (ACSDE) failed to deliver satisfactory results under noisy conditions. This is primarily due to the erroneous predictions of SVC, which further leads to inadequate initial estimates provided by population-based algorithms.

C. Evaluation of State Estimation Phase by Accuracy

In this section, we evaluate various methods in the state-estimation phase based on accuracy. First, we compared the accuracy and stability under two metrics: the
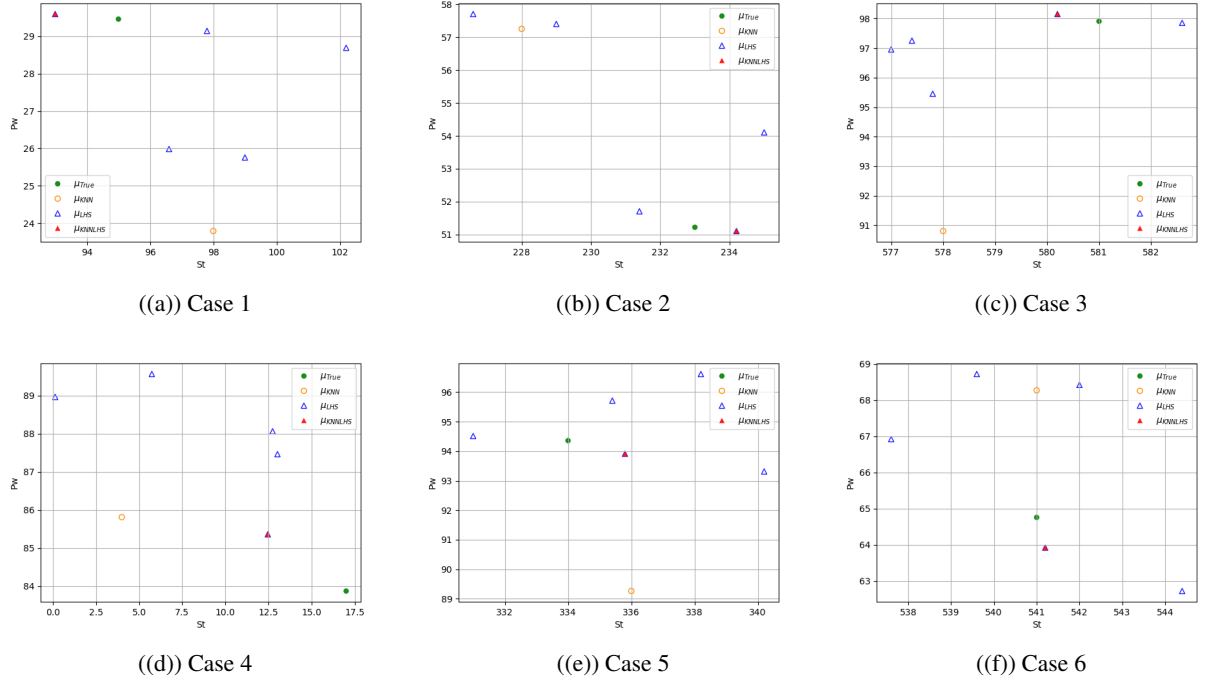
Fig. 7: Distribution of generated candidate parameters $\boldsymbol{\mu}^1 = S_t$ and $\boldsymbol{\mu}^3 = P_w$ with hybrid algorithms. The solid green points are the true values, while the hollow yellow points stand for the candidate parameter generated by the mere KNN algorithm. The hollow blue triangles represent the five optimal samplings by LHS around the initial guess given by the KNN evaluated in the observation space. The solid red points represent the output of the algorithms, which is the best among the five optimal samplings.

relative error in the L_2 norm defined by Equation 4.21 and the $L\infty$ norm defined by Equation 4.22 for field Φ, which is presented in Table 2. The 'Mean' values in the tables represent the average accuracy across 40 tested specimens, while the 'STD' values indicate the standard deviation of errors over the same set of specimens. The optimal values are indicated in bold.

In general, integration of the LHS algorithm enhanced the accuracy of the KNN model. This improvement can be attributed to the LHS sampling approach, which mitigates the discontinuity issue associated with KNN predictions by sampling around the initial estimate.

Furthermore, the NN method exhibited a decline in accuracy when confronted with noise-contaminated observational backgrounds. This is attributed to its reliance on gradients, rendering it less effective in discrete cases.

To visualize the performance of the inverse algorithms directly, we randomly selected one specimen from a test dataset. Figures (11 and 12) depict the relative L_2 error of the field prediction at the 11th vertical level. Readers are referred to Figure 10(a) for the true field values.

To fully demonstrate the reconstruction accuracy of our approach, we examined cases in which the reconstruction was relatively poor, particularly around the control rod interfaces. Reconstruction of the physical fields

in locations near the control rod interfaces is challenging because the behavior in these regions is not easily captured with high fidelity. Readers can refer to Figure 10(b) for the true field value on this axis, and Figures (13 and 14) depict the prediction errors of various methods on this axis.

However, our results also show that even in such difficult cases, our method can achieve good reconstruction in most cases. Although some techniques performed worse around the interfaces, the majority of our proposed methods still facilitated a reconstruction quality comparable to other areas across the domain. Overall, this validation indicates that the proposed approaches hold promise for representing complex reactor physics behaviors with suitable accuracy, even when challenges arise locally because of design complexities, such as control rod insertion points.

D. Evaluation of State Estimation Phase by Time Cost

In this section, we present an overview of the time costs (we take the mean of the 40 specimens) associated with each proposed algorithm in the state-estimation phase, which can be found in Table 3. The optimal val-
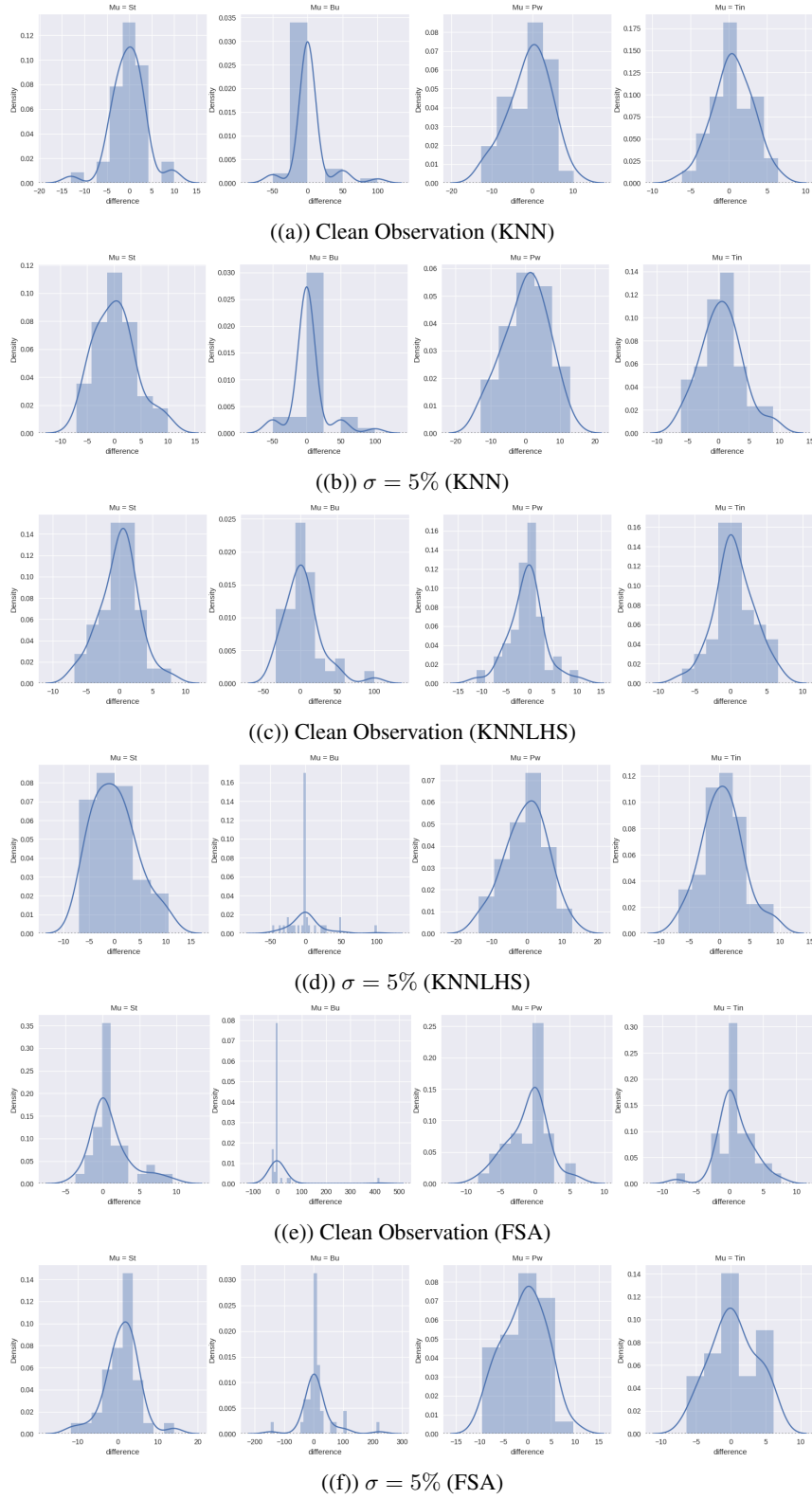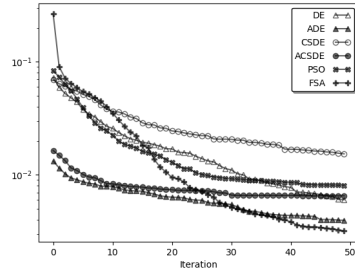
((a)) Clean Observation (KNN)

((b)) $\sigma = 5\%$ (KNN)

((c)) Clean Observation (KNNLHS)

((d)) $\sigma = 5\%$ (KNNLHS)

((e)) Clean Observation (FSA)

((f)) $\sigma = 5\%$ (FSA)

Fig. 8: Prediction deviation of μ with inverse model based on the clean or noised ($\sigma = 5\%$) observation data.

TABLE 2: Comparison of reconstructed power field errors using different algorithms.

noise level	0	0.01	0.02	0.03	0.04	0.05
			Mean L_2			
KNN	1.0662E-02	1.1128E-02	1.1461E-02	1.1461E-02	1.3296E-02	1.3541E-02
LHS	4.0457E-02	4.0210E-02	4.0823E-02	4.0754E-02	4.1138E-02	4.1790E-02
ALHS	3.4294E-02	3.4510E-02	3.4837E-02	3.4865E-02	3.5218E-02	3.7221E-02
DE	5.3219E-03	6.6446E-03	7.8485E-03	9.2407E-03	**1.1051E-02**	1.3777E-02
ADE	**3.8119E-03**	6.8077E-03	7.8185E-03	9.1585E-03	1.1134E-02	1.3526E-02
CSDE	1.7029E-02	1.6581E-02	1.7011E-02	1.7390E-02	1.8874E-02	2.0011E-02
ACSDE	7.6471E-03	4.4970E-02	4.5157E-02	4.6044E-02	4.7066E-02	4.8563E-02
PSO	1.0093E-02	1.1502E-02	1.3355E-02	1.4640E-02	1.6516E-02	1.7939E-02
FSA	3.8796E-03	**5.1561E-03**	**6.5906E-03**	**8.1864E-03**	1.0106E-02	**1.2991E-02**
NN	2.6737E-02	9.8060E-02	1.3113E-01	1.6029E-01	1.8785E-01	2.1606E-01
LHS2STEPS	3.1108E-02	3.1410E-02	3.1664E-02	3.2919E-02	3.3712E-02	3.4975E-02
KNNLHS	7.0615E-03	9.0129E-03	9.6693E-03	1.0514E-02	1.2406E-02	1.3408E-02
			STD L_2			
KNN	5.3359E-03	5.5301E-03	5.2201E-03	5.2201E-03	**5.5587E-03**	**5.7721E-03**
LHS	2.5494E-02	2.5404E-02	2.5938E-02	2.5916E-02	2.5778E-02	2.5591E-02
ALHS	2.3719E-02	2.3621E-02	2.3377E-02	2.3131E-02	2.2688E-02	2.2074E-02
DE	5.3195E-03	4.6991E-03	4.6245E-03	**4.6625E-03**	5.7638E-03	8.1052E-03
ADE	**3.1455E-03**	**4.6463E-03**	**4.5439E-03**	4.6656E-03	6.0923E-03	7.7264E-03
CSDE	9.7295E-03	7.3634E-03	7.3791E-03	7.2028E-03	7.1571E-03	7.0336E-03
ACSDE	5.2449E-03	2.2811E-02	2.4206E-02	2.4100E-02	2.3438E-02	2.2736E-02
PSO	1.2374E-02	5.3048E-03	6.9606E-03	6.2216E-03	6.6237E-03	7.2507E-03
FSA	6.2602E-03	3.5832E-03	3.5640E-03	3.7438E-03	4.6444E-03	6.0076E-03
NN	3.4667E-02	7.3152E-02	8.3294E-02	9.8081E-02	1.1251E-01	1.2140E-01
LHS2STEPS	2.1742E-02	2.1617E-02	2.1475E-02	2.1528E-02	2.1408E-02	2.0572E-02
KNNLHS	4.9637E-03	5.3025E-03	4.9676E-03	5.8222E-03	5.9288E-03	6.1111E-03
			Mean L_∞			
KNN	4.1213E-02	4.3991E-02	4.4840E-02	4.4840E-02	4.9788E-02	**4,8775E-02**
LHS	1.3126E-01	1.2567E-01	1.2326E-01	1.1998E-01	1.1904E-01	1,2479E-01
ALHS	1.0932E-01	1.1062E-01	1.1454E-01	1.1529E-01	1.1662E-01	1,2263E-01
DE	2.4302E-02	2.7539E-02	3.2492E-02	3.8239E-02	4.4672E-02	5,2324E-02
ADE	**1.5357E-02**	2.8079E-02	3.2408E-02	3.7967E-02	4.4269E-02	5,1787E-02
CSDE	5.8757E-02	6.0849E-02	6.3653E-02	6.5595E-02	6.9252E-02	7,2312E-02
ACSDE	3.1167E-02	1.4168E-01	1.4022E-01	1.4177E-01	1.4747E-01	1,4928E-01
PSO	4.6008E-02	4.6052E-02	5.2500E-02	5.6560E-02	6.2328E-02	6,7784E-02
FSA	1.6543E-02	**2.2407E-02**	**2.7837E-02**	**3.3938E-02**	**4.0081E-02**	4,9525E-02
NN	8.1834E-02	2.3886E-01	3.0443E-01	3.6030E-01	3.8488E-01	4,1158E-01
LHS2STEPS	1.0144E-01	1.0014E-01	1.0167E-01	1.0810E-01	1.1415E-01	1,1607E-01
KNNLHS	2.7072E-02	3.7594E-02	4.0961E-02	4.3185E-02	4.7107E-02	5,0309E-02
			STD L_∞			
KNN	2.7190E-02	3.0633E-02	2.8849E-02	2.8849E-02	2.8941E-02	2.9448E-02
LHS	8.7321E-02	8.2894E-02	7.8177E-02	7.6610E-02	7.7353E-02	8.1824E-02
ALHS	8.8403E-02	8.9334E-02	8.8679E-02	8.8367E-02	8.9894E-02	8.7995E-02
DE	2.2486E-02	1.8169E-02	1.9161E-02	1.9406E-02	2.4870E-02	3.2087E-02
ADE	**1.5143E-02**	1.8220E-02	1.9221E-02	2.0268E-02	2.5709E-02	3.0895E-02
CSDE	4.0726E-02	2.6558E-02	2.8031E-02	2.8521E-02	2.6937E-02	2.9098E-02
ACSDE	2.1241E-02	6.5362E-02	6.7719E-02	7.0399E-02	6.9278E-02	6.8662E-02
PSO	5.6853E-02	2.4197E-02	2.5349E-02	2.5845E-02	2.8182E-02	3.1462E-02
FSA	2.3181E-02	**1.5810E-02**	**1.5634E-02**	**1.6278E-02**	**2.0246E-02**	**2.6014E-02**
NN	9.1032E-02	1.6232E-01	1.8105E-01	1.8648E-01	2.0080E-01	1.9789E-01
LHS2STEPS	7.1699E-02	6.9753E-02	7.1235E-02	7.3554E-02	7.4717E-02	7.2363E-02
KNNLHS	2.2045E-02	2.6180E-02	2.4270E-02	2.8580E-02	2.9683E-02	3.2208E-02

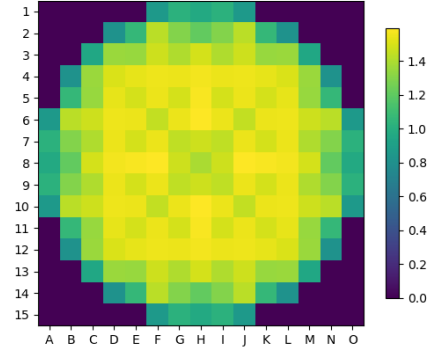((a)) Clean Observation
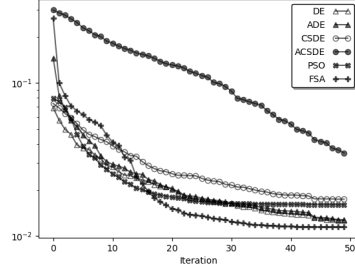
((b)) $\sigma = 1\%$
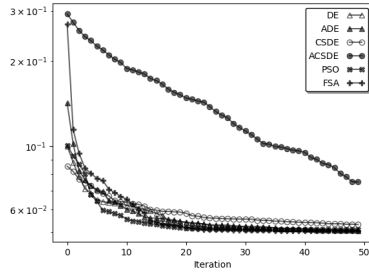
((c)) $\sigma = 5\%$

Fig. 9: Convergence rate of the cost function (defined by the L_2 norm) with different metaheuristic optimizers. The x-axis represents the iteration (generation), and we set the maximum iteration (generation) to 50.
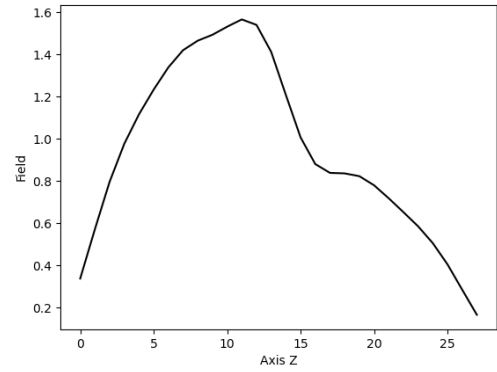

((a)) Radial power distribution of the 11th axial plane

((b)) Axial power distribution of the D01 fuel assembly

Fig. 10: True value of the power distribution on one specimen in the test dataset over the core in a realistic HPR1000 reactor.

ues are highlighted in bold, and the second optimal value is underlined.

It is worth mentioning that the NN method demonstrated notable speed. However, metaheuristic algorithms incur relatively high time costs. This is primarily attributed to the repeated computation of the cost function in Equation 2.12, which occurs 50 times for each population member over 50 iterations/generation cycles, resulting in a total of 2500 cost function evaluations. Moreover, these algorithms involve additional computations and sorting operations for information exchange and identification of the best solutions, contributing to the discrepancy in time cost.

Furthermore, it is important to note that the meth-

ods LHS, ALHS, and LHS2STEPS have approximately the same time costs. In the algorithm LHS, we set the number of samples to 1000. In algorithms ALHS and LHS2STEPS, we set the first-stage sampling number to 750, and the second-stage sampling is repeated five times, with each repetition sampling number uniformly set to 50, adding up to 1000 in total. Model-based algorithms, such as KNN and its hybrid KNNLHS, have relatively low time costs, that is, less than one second, which is acceptable in the industry.

E. Comprehensive Evaluation of Various Algorithms

In the state-estimation phase, a comparison of the proposed algorithms in terms of time cost and accuracy is presented in Figure 15. The KNNLHS algorithm demonstrates a commendable level of accuracy, achieving solutions within 0.1 s for both clean and noise-contaminated
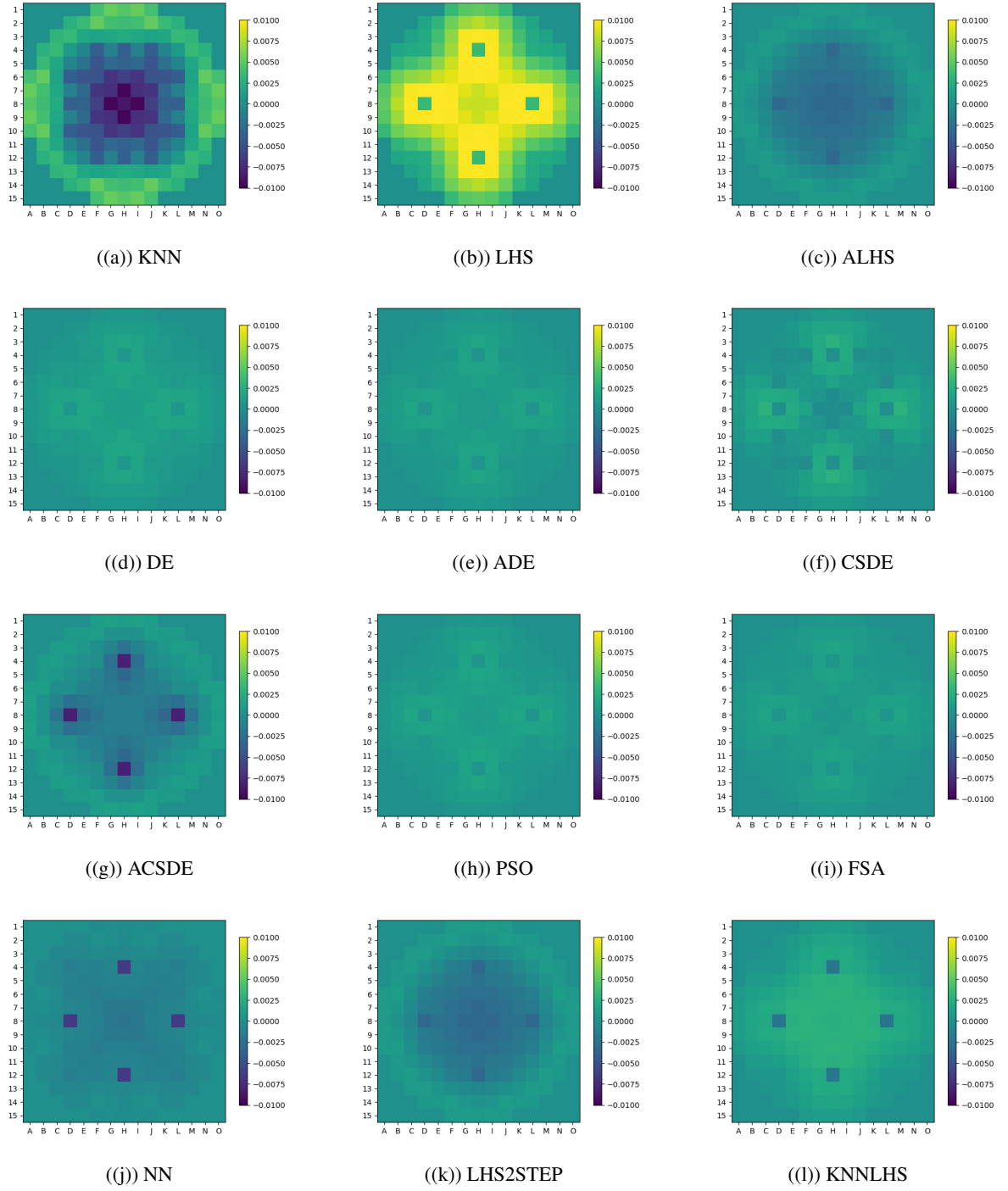
Fig. 11: Reconstructed relative errors of radial power distribution on the 11th axial plane with clean observations.

observation backgrounds. Consequently, KNNLHS can be considered suitable for real-time online inverse modeling in the context of RODT. Moreover, in scenarios in which time constraints are not limiting factors, the FSA, DE, and ADE algorithms provide reasonably accurate solutions.
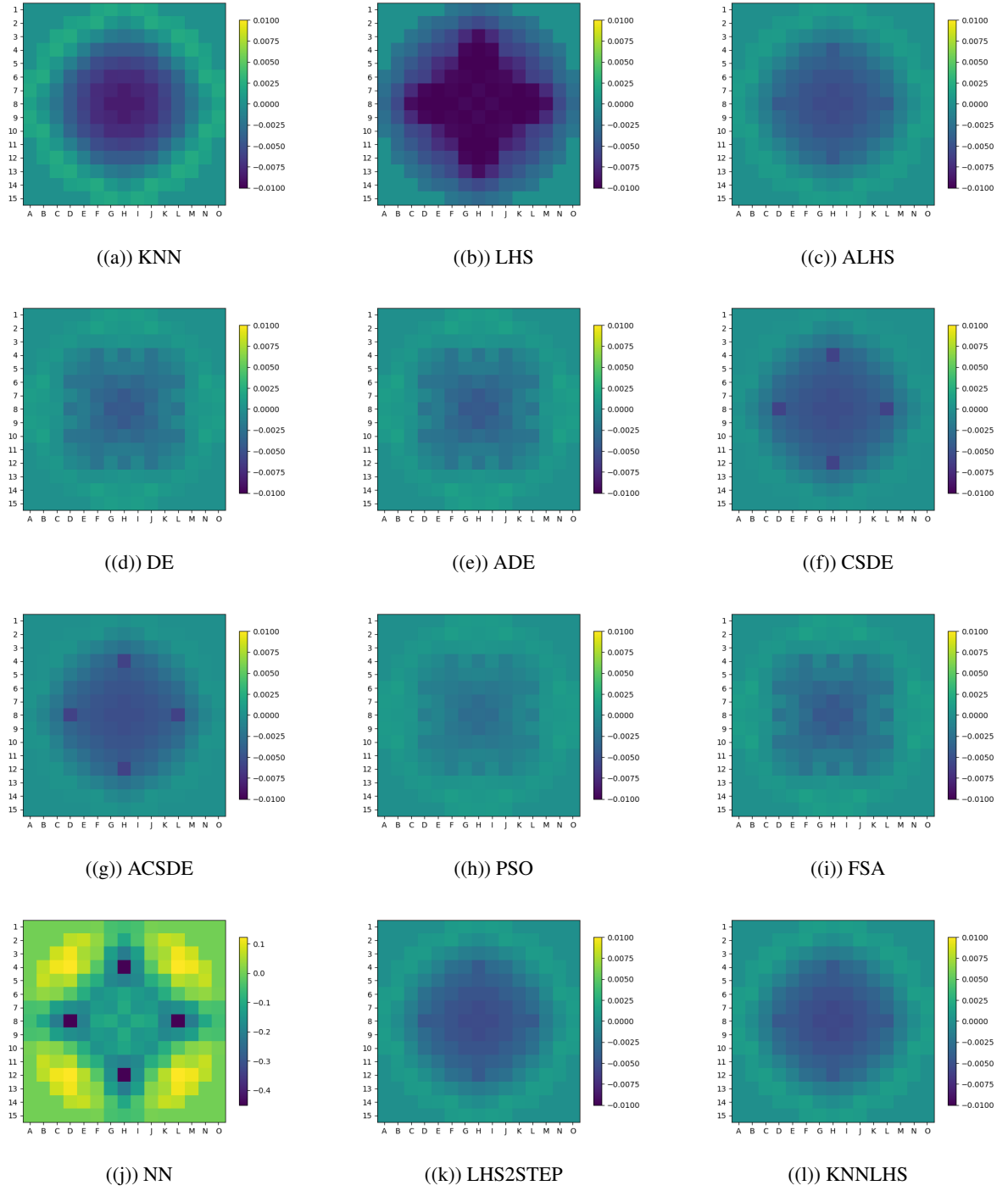
Fig. 12: Reconstructed relative errors of radial power distribution on the 11th axial plane with observations of noise level $\sigma = 5\%$.

VI. CONCLUSION AND FUTURE WORKS

In this study, we developed and implemented a series of powerful algorithms for the RODT, aiming to address the challenges in parameter identification and state estimation for nuclear reactor monitoring and optimization applications. Novel hybrid optimization algorithms, including ADE, CSDE, ACSDE, and KNNLHS, were de-
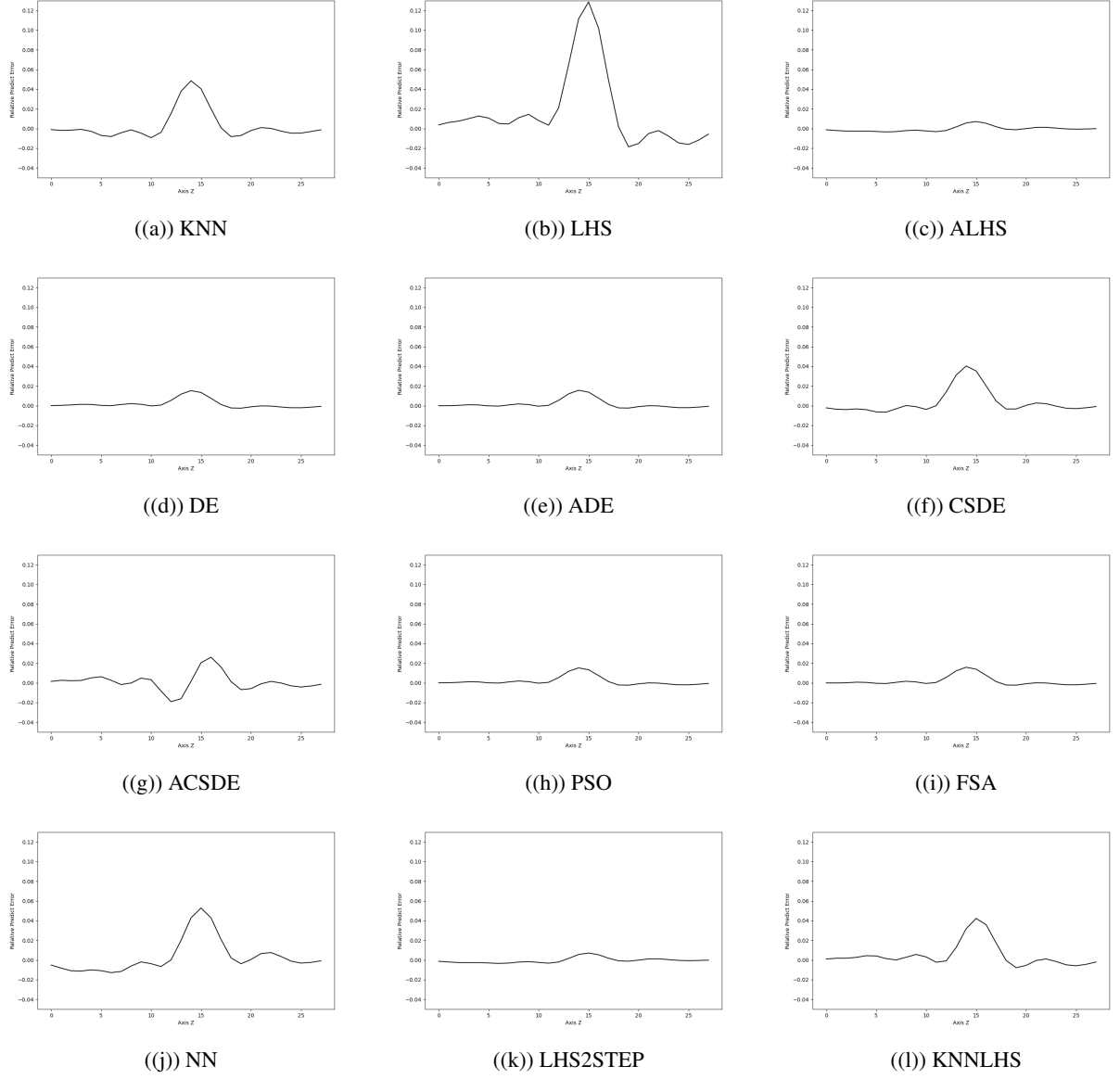
Fig. 13: Predicted relative error for the D01 fuel assembly with clean observations.

TABLE 3: Comparison of time cost for different algorithms.

| noise level	0	0.01	0.02	0.03	0.04	0.05
KNN	6.3696E-03	3.1396E-03	3.1599E-03	3.1594E-03	3.1915E-03	3.1527E-03
LHS	7.3946E-01	7.3326E-01	7.2510E-01	7.3279E-01	7.2927E-01	7.2180E-01
ALHS	7.1437E-01	7.3674E-01	7.1443E-01	7.1613E-01	7.1730E-01	7.1050E-01
DE	3.0745E+00	2.9459E+00	2.8510E+00	2.9223E+00	2.8299E+00	2.7874E+00
ADE	2.9249E+00	2.9339E+00	2.8915E+00	2.9284E+00	2.9743E+00	3.0290E+00
CSDE	4.7069E+00	4.6679E+00	4.6691E+00	4.6531E+00	4.6615E+00	4.6227E+00
ACSDE	4.8054E+00	4.6563E+00	4.6374E+00	4.6886E+00	4.6303E+00	4.7069E+00
PSO	2.4404E+00	2.5169E+00	2.4969E+00	2.3198E+00	2.3551E+00	2.3151E+00
FSA	2.7848E+00	2.8079E+00	2.7324E+00	2.7212E+00	2.7090E+00	2.6977E+00
NN	**4.1338E-04**	**4.2277E-04**	**4.0678E-04**	**4.0900E-04**	**4.0602E-04**	**4.0516E-04**
LHS2STEPS	7.0137E-01	7.0546E-01	7.0171E-01	7.0508E-01	7.0827E-01	7.0630E-01
KNNLHS	7.0515E-02	6.4439E-02	6.3790E-02	6.4109E-02	6.3136E-02	6.3097E-02
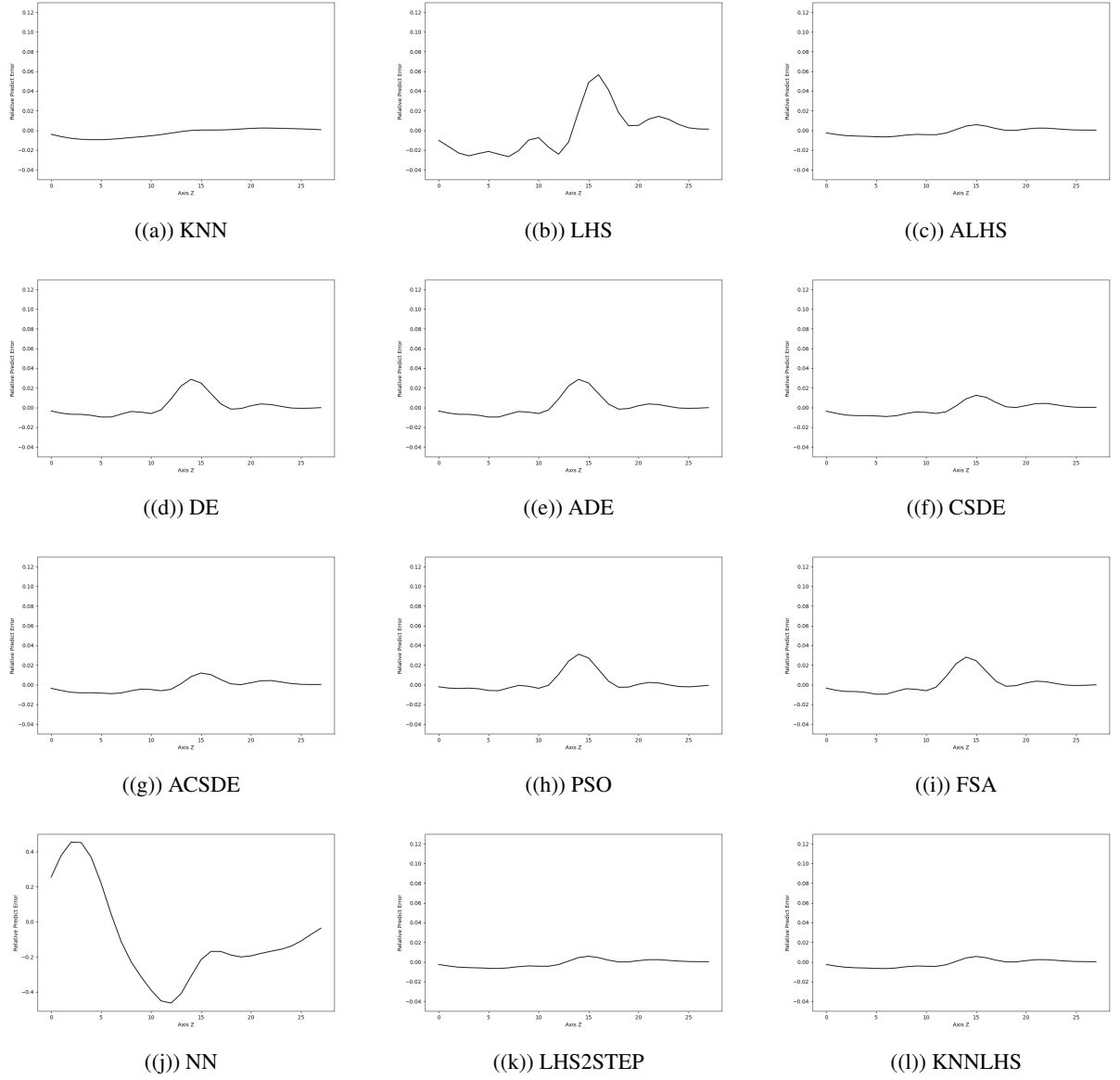
Fig. 14: Predicted relative error for the D01 fuel assembly with observations of noise level $\sigma = 5\%$.

signed and implemented to effectively solve the inverse problems posed by the discontinuous surrogate forward modeling approach. Both deterministic and metaheuristic optimization methods have been investigated for their applicability to nuclear engineering systems characterized by high-dimensional, nonlinear relationships. The integration of coarse and fine grid searching strikes an optimal balance between computational efficiency and solution accuracy.

Through extensive experimentation and evaluation, we conducted a thorough comparative analysis of parameter identification accuracy, field reconstruction accuracy, and convergence profile. FSA exhibited the fastest

convergence rate, and the proposed ADE method follows closely behind. Moreover, integrating standard DE or CSDE algorithms within the SVC coarse grid search framework enables these techniques to effectively escape local optima in early iterations while maintaining relatively low-cost function values, surpassing their standalone implementations. Additionally, it is observed that pairing the LHS approach within the KNN surrogate model enhances the accuracy by mitigating discontinuity issues through local sampling around the initial guess. Thus, our approach effectively addresses the challenges of discontinuity and non-differentiability in surrogate forward mapping generated by the KNN model.
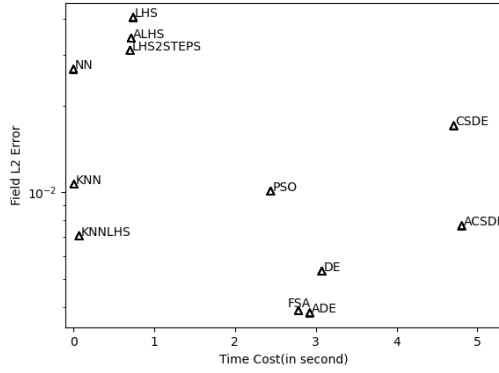
((a)) Cases with clean observations
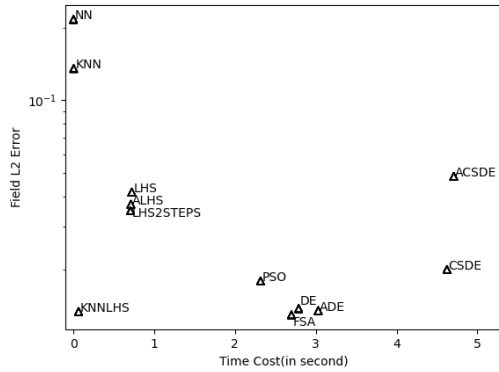
((b)) Cases with observations of noise level $\sigma = 5\%$

Fig. 15: Comparison of the synthetic performance for different inverse algorithms.

Conversely, the NN method experiences declining accuracy when faced with noise-contaminated measurements, likely owing to overreliance on gradient information, which proves less effective under such uncertainty.

Our algorithms demonstrated high computational efficiency, with solutions obtained within 1 s and as fast as ≤ 0.08 s for KNN, KNNLHS, and NN. They also exhibited robustness in handling noise-contaminated backgrounds. Our work contributes to the realization of RODT by providing effective, efficient, and reliable support for real-world nuclear power plant operation and sensor data. These algorithms establish a promising foundation for advanced system modeling and engineering decision support.

By specifically tailoring solutions to nuclear inverse and surrogate modeling challenges, this research advances the state-of-the-art RODT methodology. The improved parameter identification and state estimation facilitate the optimization of reactor monitoring, control, and performance through data-driven DT applications. In the future, novel methods can contribute to our RODT system, and with the same methodology, we can support the development and implementation of these transformative technologies in other industries requiring real-time DTs.

Continued work validating diverse systems and progressively integrating multi-physics modeling holds promise for fully realizing the potential of RODTs in transforming nuclear engineering practice with powerful real-time and decision-support capabilities. Realizing this potential requires ongoing progress, collaboration, and open innovation across expert areas.

COMPETING INTERESTS

The authors declare no conflict of interest.

[1] F. Tao, H. Zhang, A. Liu, et al., Digital twin in industry: State-of-the-art. IEEE Transactions on industrial informatics **15**, 2405–2415 (2018).

[2] M.W. Grieves, Virtually intelligent product systems: Digital and physical twins. (, 2019)

[3] K.Y.H. Lim, P. Zheng, C.H. Chen, A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. J. Intell. Manuf. **31**, 1313–1337 (2020).

[4] M.W. Grieves, J.H. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. 2017

[5] M. Liu, S. Fang, H. Dong, et al., Review of digital twin about concepts, technologies, and industrial applications. J. Manuf. Syst. **58**, 346–361 (2021).

[6] M. Singh, E. Fuenmayor, E.P. Hinchy, et al., Digital twin: Origin to future. Appl. Syst. Innov. **4**, 36 (2021).

[7] E. Örs, R. Schmidt, M. Mighani, et al., A conceptual framework for ai-based operational digital twin in chemical process engineering. 2020 IEEE Int. Conf. on Eng. Technol. Innov. (ICE/ITMC) 1–8 (2020).

[8] J. Kraft, S. Kuntzagk, Engine fleet-management: The use of digital twins from a mro perspective. 2017

[9] F. Chinesta, E. Cueto, E. Abisset-Chavanne, et al., Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data. Arch. Comput. Methods Eng. **27**, 105–134 (2018).

[10] M. Adams, X. Li, L. Boucinha, et al., Hybrid digital twins: A primer on combining physics-based and data analytics approaches. IEEE Softw. **39**, 47–52 (2022). doi:10.1109/MS.2021.3134042

[11] H. Song, M. Song, X. Liu, Online autonomous calibration of digital twins using machine learning with application to nuclear power plants. Appl. Energy **326**, 119995 (2022). doi:https://doi.org/10.1016/j.apenergy.2022.119995

[12] A.K. Sleiti, J.S. Kapat, L. Vesely, Digital twin in energy industry: Proposed robust digital twin for power plant and other complex capital-intensive large engineering systems. Energy Reports **8**, 3704–3726 (2022). doi:https://doi.org/10.1016/j.egyr.2022.02.305

[13] H. Gong, S. Cheng, Z. Chen, et al., Data-enabled physics-informed machine learning for reduced-order modeling digital twin: Application to nuclear reactor physics. Nucl. Sci. Eng. **196**, 1–26 (2022). doi:10.1080/00295639.2021.2014752

[14] X. Li, S. Wang, W. Zhou, et al., in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Research on fault diagnosis algorithm based on convolutional neural network. Vol. 1, 2019, pp. 8–12. doi:10.1109/IHMSC.2019.00010

[15] H. Gong, S. Cheng, Z. Chen, et al., An efficient digital twin based on machine learning svd autoencoder and generalised latent assimilation for nuclear reactor physics. Annals Nucl. Energy **179**, 109431 (2022). doi:10.1016/j.anucene.2022.109431

[16] H. Gong, T. Zhu, Z. Chen, et al., Parameter identification and state estimation for nuclear reactor operation digital twin. Annals Nucl. Energy **180**, 610041 (2023). doi:10.1016/j.anucene.2022.109497

[17] A.C. Antoulas, D.C. Sorensen, S. Gugercin, A survey of model reduction methods for large-scale systems. 2000

[18] R. Arcucci, L. Mottet, C. Pain, et al., Optimal reduced space for variational data assimilation. J. Comput. Phys. **379**,. doi:10.1016/j.jcp.2018.10.042

[19] J.P. Argaud, B. Bouriquet, F. de Caso, et al., Sensor placement in nuclear reactors based on the generalized empirical interpolation method. J. Comput. Phys. **363**, 354–370 (2018).

[20] P. Benner, M. Ohlberger, A. Cohen, et al., *Model reduction and approximation: theory and algorithms*, (SIAM, 2017)

[21] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. **57**, 483–531 (2015). arXiv:https://doi.org/10.1137/130932715, doi:10.1137/130932715

[22] M.G. Kapteyn, D.J. Knezevic, K.E. Willcox, Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. 2020

[23] D. Hartmann, M. Herz, U. Wever, *Model Order Reduction a Key Technology for Digital Twins*, (Springer International Publishing, Cham, 2018), pp. 167–179. doi:10.1007/978-3-319-75319-5_8

[24] J. Hammond, R. Chakir, F. Bourquin, et al., Pbdw: A non-intrusive reduced basis data assimilation method and its application to an urban dispersion modeling framework. Appl. Math. Model. **76**, 1–25 (2019). doi:https://doi.org/10.1016/j.apm.2019.05.012

[25] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access **8**, 21980–22012 (2020). doi:10.1109/ACCESS.2020.2970143

[26] E. Nadal, F. Chinesta, P. Díez, et al., Real time parameter identification and solution reconstruction from experimental data using the proper generalized decomposition. Comput. Methods Appl. Mech. Eng. **296**, 113–128 (2015). doi:https://doi.org/10.1016/j.cma.2015.07.020

[27] V. Harish, A. Kumar, Reduced order modeling and parameter identification of a building energy system model through an optimization routine. Appl. Energy **162**, 1010–1023 (2016). doi:https://doi.org/10.1016/j.apenergy.2015.10.137

[28] H. Fu, H. Wang, Z. Wang, Pod/deim reduced-order modeling of time-fractional partial differential equations with applications in parameter identification. J. Sci. Comput. **74**, 220–243 (2016).

[29] Q. Ding, Y. Wang, Z. Chen, Parameter identification of reduced-order electrochemical model simplified by spectral methods and state estimation based on square-root cubature kalman filter. J. Energy Storage **46**, 103828 (2022). doi:https://doi.org/10.1016/j.est.2021.103828

[30] Y. Maday, O. Mula, A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation. Springer INdAM Ser. **4**,. doi:10.1007/978-88-470-2592-9_13

[31] Y. Maday, A.T. Patera, J.D. Penn, et al., A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. Int. J. for Numer. Methods Eng. **102**, 933–965 . arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4747, doi:https://doi.org/10.1002/nme.4747

[32] P. Binev, A. Cohen, W. Dahmen, et al., Data assimilation in reduced modeling. SIAM/ASA J. Uncertain. Quantification **5**, 1–29 (2015).

[33] H. Gong, Data assimilation with reduced basis and noisy measurement: Applications to nuclear reactor cores. Ph.D. thesis, Sorbonne université (2018)

[34] R. Arcucci, L. Mottet, C. Pain, et al., Optimal reduced space for variational data assimilation. J. Comput. Phys. **379**, 51–69 (2019). doi:https://doi.org/10.1016/j.jcp.2018.10.042

[35] H. Gong, Z. Chen, Y. Maday, et al., Optimal and fast field reconstruction with reduced basis and limited observations: Application to reactor core online monitoring. Nucl. Eng. Des. **377**, 111113 (2021). doi:https://doi.org/10.1016/j.nucengdes.2021.111113

[36] F. Di Rocco, D.G. Cacuci, Sensitivity and uncertainty analysis of a reduced-order model of nonlinear bwr dynamics: I. forward sensitivity analysis. Annals Nucl. Energy **148**, 107738 (2020). doi:https://doi.org/10.1016/j.anucene.2020.107738

[37] S. Peitz, S. Ober-Blöbaum, M. Dellnitz, Multiobjective optimal control methods for the navier-stokes equations using reduced order modeling. Acta Appl. Math. **161**, 171–199 (2018).

[38] P. Chen, A. Quarteroni, G. Rozza, Reduced order methods for uncertainty quantification problems. ETH Zurich, SAM Rep. **3**,.

25

[39] Y. Liu, X. Sun, N.T. Dinh, Validation and uncertainty quantification of multiphase-cfd solvers: A data-driven bayesian framework supported by high-resolution experiments. Nucl. Eng. Des. **354**, 110200 (2019). Special Issue on TRENDS AND PERSPECTIVES IN NUCLEAR THERMAL-HYDRAULICS. doi:https://doi.org/10.1016/j.nucengdes.2019.110200

[40] M. Braun, Reduced order modelling and uncertainty propagation applied to water distribution networks. 2019

[41] G. Carere, M. Strazzullo, F. Ballarin, et al., A weighted pod-reduction approach for parametrized pde-constrained optimal control problems with random inputs and applications to environmental sciences. Comput. Math. Appl. **102**, 261–276 (2021).

[42] N. Demo, G. Ortali, G. Gustin, et al., An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques. Bollettino dell'Unione Matematica Italiana **14**, 211–230 (2020).

[43] S.A. Renganathan, Koopman-based approach to nonintrusive reduced order modeling: Application to aerodynamic shape optimization and uncertainty propagation. AIAA J. **58**, 2221–2235 (2020).

[44] N.T. Mücke, L.H. Christiansen, A.P. Engsig-Karup, et al., Reduced order modeling for nonlinear pde-constrained optimization using neural networks. 2019 IEEE 58th Conf. on Decis. Control. (CDC) 4267–4272 (2019).

[45] M. Heinkenschloss, D. Jando, Reduced order modeling for time-dependent optimization problems with initial value controls. SIAM J. Sci. Comput. **40**,.

[46] J.V. Aguado, D. Borzacchiello, C. Ghnatios, et al., A simulation app based on reduced order modeling for manufacturing optimization of composite outlet guide vanes. Adv. Model. Simul. Eng. Sci. **4**, 1–26 (2017).

[47] L. Iapichino, S. Ulbrich, S. Volkwein, Multiobjective pde-constrained optimization using the reduced-basis method. Adv. Comput. Math. **43**, 945–972 (2017).

[48] M.L. Zhang, Z.H. Zhou, Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognit. **40**, 2038–2048 (2007).

[49] X. Wu, V. Kumar, J.R. Quinlan, et al., Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**, 1–37 (2007).

[50] S. Zhang, X. Li, M. Zong, et al., Learning k for knn classification. ACM Trans. Intell. Syst. Technol. **8**,. doi:10.1145/2990508

[51] A. Tarantola, Inverse problem theory - and methods for model parameter estimation. 2004

[52] M. Frangos, Y. Marzouk, K. Willcox, et al., Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems. Large-Scale Inverse Probl. Quantification Uncertain. 123–149 (2010).

[53] P. An, Y. Ma, P. Xiao, et al., Development and validation of reactor nuclear design code corca-3d. Nucl. Eng. Technol. **51**, 1721–1728 (2019). doi:https://doi.org/10.1016/j.net.2019.05.015

[54] N. El-Sahlamy, M. Hassan, A. Khedr, et al., Study of rod ejection accident at hot zero power condition in a pwr using relap5. Prog. Nucl. Energy **144**, 104100 (2022). doi:https://doi.org/10.1016/j.pnucene.2021.104100

[55] Y. Maday, Reduced basis method for the rapid and reliable solution of partial differential equations. Proc. oh

[56] M.A. Grepl, Y. Maday, N.C. Nguyen, et al., Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. ESAIM: Math. Model. Numer. Analysis **41**, 575–605 (2007).

[57] J.S. Hesthaven, G. Rozza, B. Stamm, et al., *Certified reduced basis methods for parametrized partial differential equations*, Vol. 590, (Springer, 2016)

[58] C. Eckart, G.M. Young, The approximation of one matrix by another of lower rank. Psychom. **1**, 211–218 (1936).

[59] S. Arridge, P. Maass, O. Öktem, et al., Solving inverse problems using data-driven models. Acta Numer. **28**, 1–174 (2019).

[60] R. Ahuja, J. Orlin, Inverse optimization. Oper. Res. **49**,. doi:10.1287/opre.49.5.771.10607

[61] R.M. Lewis, V. Torczon, M.W. Trosset, Direct search methods: then and now. J. Comput. Appl. Math. **124**, 191–207 (2000).

[62] J.C. Helton, F.J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab. Eng. & Syst. Saf. **81**, 23–69 (2003).

[63] R. Biedrzycki, J. Arabas, D. Jagodziński, Bound constraints handling in differential evolution: An experimental study. Swarm Evol. Comput. **50**, 100453 (2019).

[64] N.D. Lagaros, M. Kournoutos, N.A. Kallioras, et al., Constraint handling techniques for metaheuristics: a state-of-the-art review and new variants. Optim. Eng. 1–48 (2023).

[65] M. Montemurro, A. Vincenti, P. Vannucci, The automatic dynamic penalisation method (adp) for handling constraints with genetic algorithms. Comput. Methods Appl. Mech. Eng. **256**, 70–87 (2013).

[66] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evol. Comput. **15**, 4–31 (2011). doi:10.1109/TEVC.2010.2059031

[67] R. Storn, K.V. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997).

[68] R. Eberhart, J. Kennedy, in *Proceedings of the IEEE international conference on neural networks*, Particle swarm optimization. Vol. 4, Citeseer, 1995, pp. 1942–1948

[69] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis Mach. Intell. -6, 721–741 (1984). doi:10.1109/TPAMI.1984.4767596

[70] S. Kirkpatrick, C. Gelatt, M. Vecchi, in *Readings in Computer Vision*, Optimization by simulated annealing. (Morgan Kaufmann, San Francisco (CA), 1987), pp. 606–615 doi:https://doi.org/10.1016/B978-0-08-051581-6.50059-3

[71] T. Guilmeau, E. Chouzenoux, V. Elvira, in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, Simulated annealing: a review and a new scheme. 2021, pp. 101–105. doi:10.1109/SSP49050.2021.9513782

[72] V.P. Tran, G.T. Phan, V.K. Hoang, et al., Evolutionary simulated annealing for fuel loading optimization of vver-1000 reactor. Annals Nucl. Energy **151**, 107938 (2021). doi:https://doi.org/10.1016/j.anucene.2020.107938

[73] H. Szu, R. Hartley, Fast simulated annealing. Phys. Lett. A **122**, 721–741 (1984).

Int. Congr. Math. Vol. 3, 2006-01-01, ISBN 978-3-03719-022-7, pags. 1255-1270 **3**,.

[74] X.S. Yang, S. Deb, in *2009 World congress on nature & biologically inspired computing (NaBIC)*, Cuckoo search via lévy flights. Ieee, 2009, pp. 210–214

[75] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: A tutorial. Comput. **29**, 31–44 (1996).

[76] J. Brajard, A. Carrassi, M. Bocquet, et al., Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the lorenz 96 model. J. Comput. Sci. **44**, 101171 (2020). doi:https://doi.org/10.1016/j.jocs.2020.101171

[77] Y. Yang, H. Gong, S. Zhang, et al., A data-enabled physics-informed neural network with comprehensive numerical study on solving neutron diffusion eigenvalue problems. Annals Nucl. Energy **183**, 109656 (2023). doi:https://doi.org/10.1016/j.anucene.2022.109656

[78] M. Salah, J. Trinder, Support vector machines: Optimization and validation for land cover mapping using aerial images and lidar data. .

[79] B. Zhang, G. Jin, J. Zhu, Towards automatic freeform optics design: coarse and fine search of the three-mirror solution space. Light. Sci. & Appl. **10**,.

[80] J. Bergstra, Y. Bengio, Random search for hyperparameter optimization. J. machine learning research **13**,.

[81] X.S. Yang, *Nature-Inspired Metaheuristic Algorithms*, (, 2010)

[82] D. Leon Valido, A. Gonzalez, Mutations as levy flights. Sci. Reports **11**,. doi:10.1038/s41598-021-88012-1

[83] A. Luque, A. Carrasco, A. Martín, et al., The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognit. **91**, 216–231 (2019). doi:https://doi.org/10.1016/j.patcog.2019.02.023

[84] C. Goutte, E. Gaussier, in *Advances in Information Retrieval*, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005), pp. 345–359

[85] X. Li, Q. Liu, Q. Li, et al., 177 core nuclear design for hpr1000. Nucl. Power Eng. **40**, 0008–00012 (2019).

[86] P. An, Y. Ma, P. Xiao, et al., Development and validation of reactor nuclear design code corca-3d. Nucl. Eng. Technol. **51**, 1721–1728 (2019).